GUIDELINES FOR COMPUTER BRAILLE CODE

PRESENTATION:
28TH ANNUAL CTEVH CONFERENCE

BY: ELINOR SAVAGE, CTEVH SPECIALIST
COMPUTER BRAILLE NOTATION

HV1689
.S38
1987

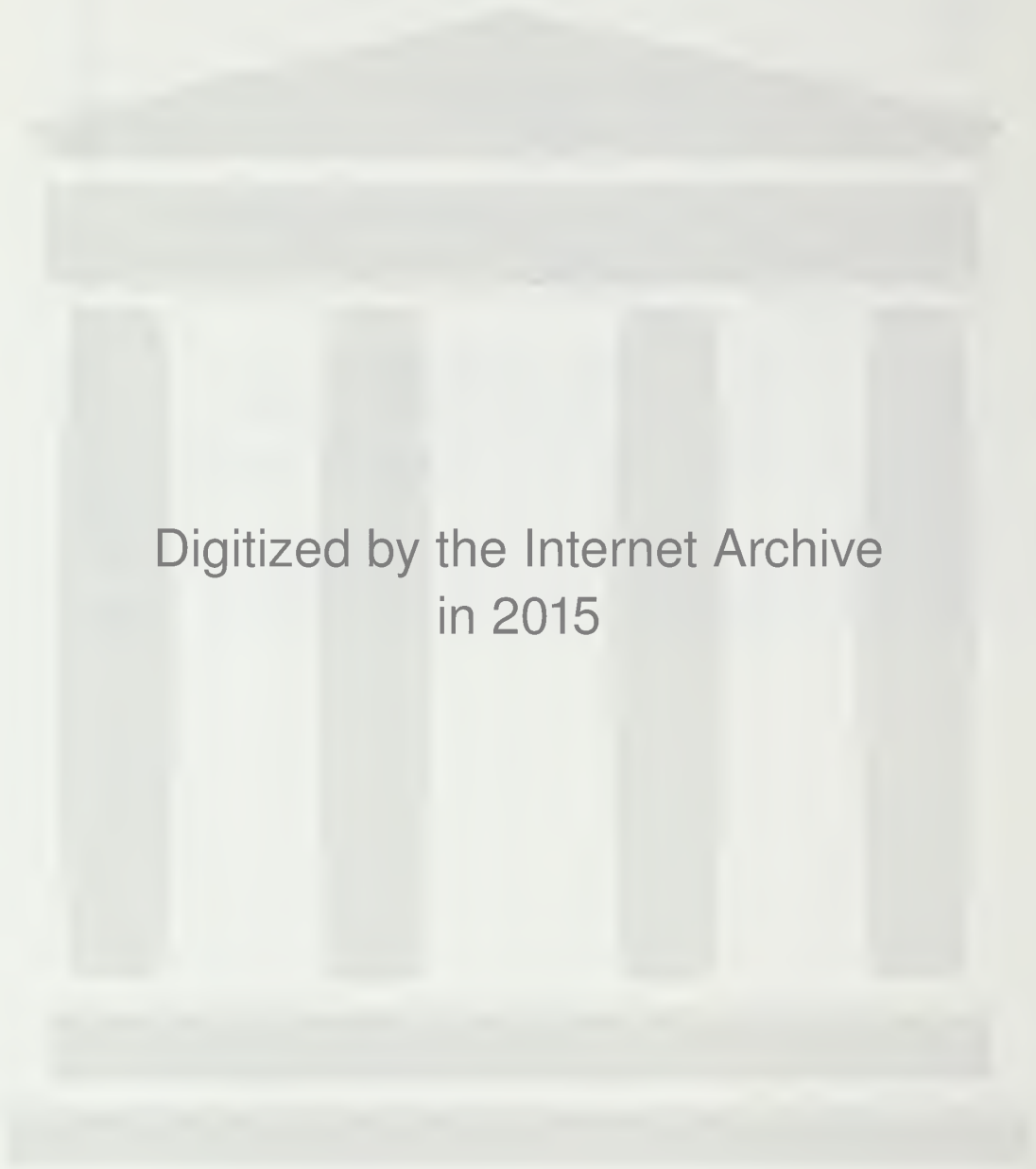
COMPUTER BRAILLE CODE

CTEVH WORKSHOP: March 19, 1987

Chairman: Joyce Van Tuyl
Presentation: Elinor Savage
Consultant: Priscilla Harris
Braille: Carolyn Card

CONTENTS

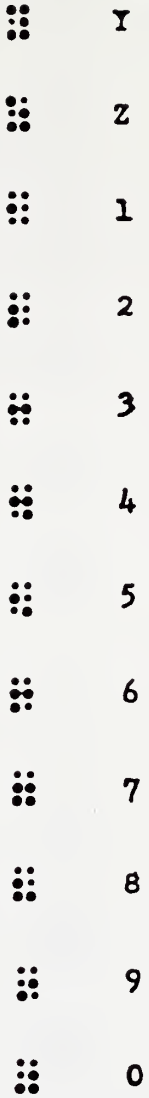
	Computer Braille Code Symbols	i
	Introduction	1
I.	Guidelines For Identifying Computer Notation	1
II.	Notation in the Primary Code	10
III.	Indicators	25
IV.	Shift Indicator	26
V.	Caps Lock and Caps Release Indicators	28
VI.	Emphasis Indicators	37
VII.	Shape Indicators	45
VIII.	Nemeth Code Indicators	52
IX.	Computer Braille Code Indicators	58
X.	Half-Line Shift Indicators	72
XI.	Transcriber's Option Symbols	76
XII.	Termination Indicator	81
XIII.	Summary of the Rules for the Use of the End and Termination Indicators	81
XIV.	Countable Spaces Indicator	89
XV.	Continuation Indicator	92
XVI.	Isolated Lower-Cell Sign	93
XVII.	Representation of Control Characters	94
XVIII.	Summary of Rules for Computer Code Notation	95
XIX.	Summary of Format	99
XX.	Displayed Computer Notation: Programs--Format	99
XXI.	Comments	106
XXII.	Indentation	114
XXIII.	Structuring The Text	115
XXIV.	Suggestions for Screen Displays, Flowcharts, and Syntax Diagrams	119



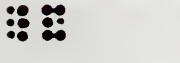
Digitized by the Internet Archive
in 2015

https://archive.org/details/guidelinesforcom00byel

COMPUTER BRAILLE CODE



Print	Braille	Meaning	Print	Braille	Meaning
.		period	@		at
,		comma	*		asterisk
:		colon	/		slash
;		semicolon	|		vertical bar
?		question mark	\		backslash
!		exclamation point	[		left bracket
"		quotation marks	]		right bracket
'		apostrophe	{		left brace
-		hyphen, minus sign	}		right brace
+		plus	_		underscore
(		left parenthesis	↑ or ^		up arrow/caret
)		right parenthesis	`		accent grave
#		number sign	~		tilde
$		dollar sign	=		equals
%		percent sign	<		less than
&		ampersand	>		greater than

Braille	Meaning
	caps lock indicator
	caps release indicator
	begin emphasis
	end emphasis
	continuation indicator
	countable spaces indicator
	shift/lower-cell sign indicator
	transcriber's primary option symbol
	transcriber's secondary option symbol
	begin Computer Braille Code indicator
	begin Nemeth Braille Code indicator
	begin shape indicator
	half-line shift up indicator
	half-line shift down indicator
	end Computer/Nemeth/shape/half-line shifts indicator

INTRODUCTION

The Computer Braille Code, adopted 1986, is a completely new code for transcribing computer-related materials. A few of the symbols and rules are like those of the previous "provisional" code, but most are new and must be learned just as any braille code.

In this presentation the rules of the Computer Braille Code have been restated or expanded where deemed necessary; examples in the Code are referred to in sections where they apply and additional examples are given; suggestions for handling some situations not covered by the Code are given.

You are urged to submit questions, comments, examples, etc. of computer material you encounter, to assist in the refinement of this code. Always include a photocopy of the material in question when you send an inquiry to your specialist. Occasionally the specialist may need to see the entire book, so be sure to include the complete title, author, copyright date and publisher with the photocopy.

1. GUIDELINES FOR IDENTIFYING COMPUTER NOTATION (Section 1 - CBC)

Materials that should be transcribed in the Computer Braille Code include:

1. Manuals which accompany computers and/or computer programs and which document the way in which the computer or program is used. For example,

   "APPLE II DOS Programmer's Manual"
   "Executive Filer - Software for the IBM PC"

2. Textbooks relating to computer mechanisms or computer languages. For example,

   "Computer Networks"
   "Problem Solving and Structural Programming in FORTRAN"
   "Introduction to Data Structure"
   "C Programming Guide"

3. Computer programs or other material identified as computer notation in mathematics textbooks. For example,

   Heading in an algebra book - "Computer Key-In"

4. Other specifically designated computer materials. For example,

   "Quick Reference Card - IBM Assistant Series"
   "CompuServe User's Guide"

Most of the manuals and textbooks being printed today will include a description at the beginning of the book of the type styles used for computer notation.

1.

Conventions

Throughout this manual, the following conventions are used:

UPPER CASE denotes the item must be entered (typed) exactly as shown.

lower case denotes the entry of a variable that can be one of many possible responses; accompanying text informs you of the type of entry to be made.

[ ] denotes an optional entry. If multiple items are stacked within brackets, you may enter one of them or none. For example, **TYP[E]** means that you must enter **TYP** but the E is not required.

{ } denotes a required entry chosen from among the included items.

␣ denotes the entry of a space. This symbol is used where needed for clarity only.

(ENTER) denotes depressing the ENTER key (carriage return on some keyboards). This symbol is used where needed for clarity only. Unless otherwise specified, all user entries are terminated with an ENTER key (or carriage return).

^C denotes entry of a Control-C (control key is held down while the C is depressed). Other control characters are similarly illustrated.

Blue in examples indicates input from you as opposed to output from the computer.

2.

A Session with UNIX

Let's begin with an annotated dialog between you and your UNIX system. Throughout the examples in this book, what you type is printed in *slanted letters*, computer responses are in typewriter-style characters, and explanations are in *italics*.

Establish a connection: dial a phone or turn on a switch as necessary.
Your system should say

```
login: you                    Type your name, then press RETURN
Password:                     Your password won't be echoed as you type it
You have mail.                There's mail to be read after you log in
$                             The system is now ready for your commands
$                             Press RETURN a couple of times
$ date                        What's the date and time?
Sun Sep 25 23:02:57 EDT 1983
```

3.

To increase legibility, the programs, examples, C keywords, and variables are set in a font called OCR-B. This font is reproduced below

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789
!"#$%&'()*+,-./
:;<=>?@[]\^_{|}~
```

ADDITIONAL EXAMPLES OF COMPUTER NOTATION

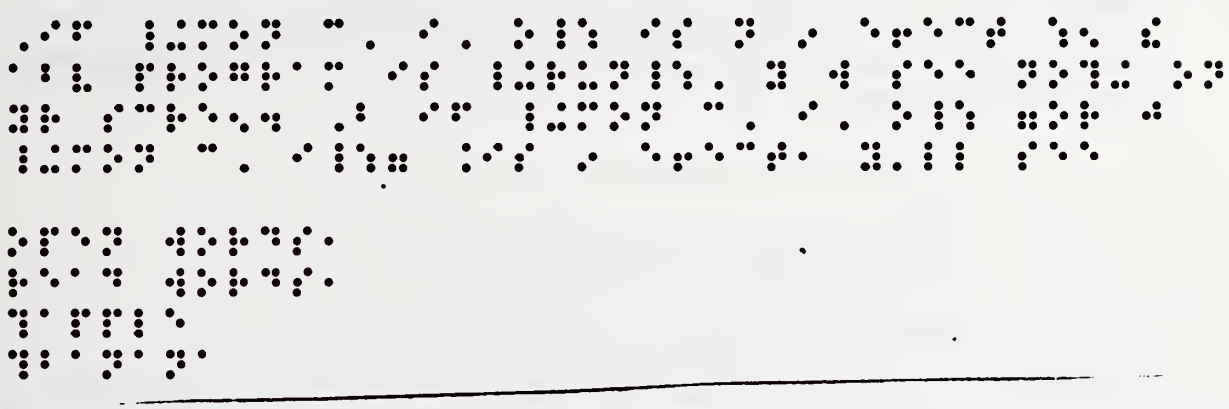
NOTE: All examples in this presentation in which only upper-case letters comprise computer notation may be transcribed either with the Caps Lock Indicators or the Computer Braille Code Indicators, depending on whether upper-case is or is not used exclusively for such notation in the text. This decision has to be made when the text is structured. (See Section XXIII.)

1. Programs are all in green type. Computer notation in this book is exclusively upper-case.

If MON C, I, O is not in effect when the above program is RUN, you will see nothing on your screen. But if MON C, I, O (or just MON C, I) is in effect, you'll see

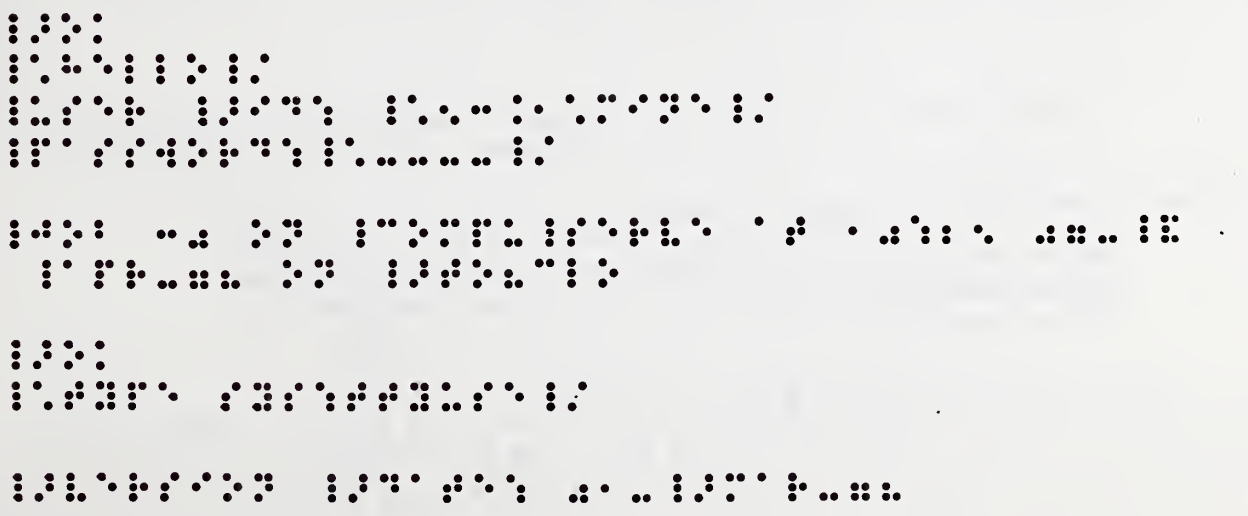
```
OPEN WORDS1
READ WORDS1
?APPLE
?BANANA
```

[green type]



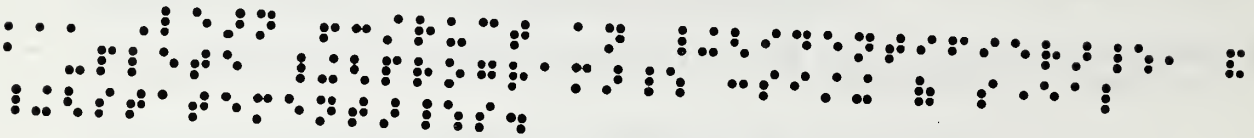
2. Underlining is used in text.

```
OK
nello
User ID: 53.1*mine
Password:

Job 30 on CompuServe at 10:25 07-Apr-78 on T08CL0

OK
type sys:ttyuse

VERSION DATE: 01-MAR-78
```



3. Words enclosed in angle brackets

1d. Learn to use syntax diagrams as concise representations of the rules for constructing programs using the Pascal programming language. [Learn to construct an <identifier>, and a complete <program> consisting of several <statement>s.]

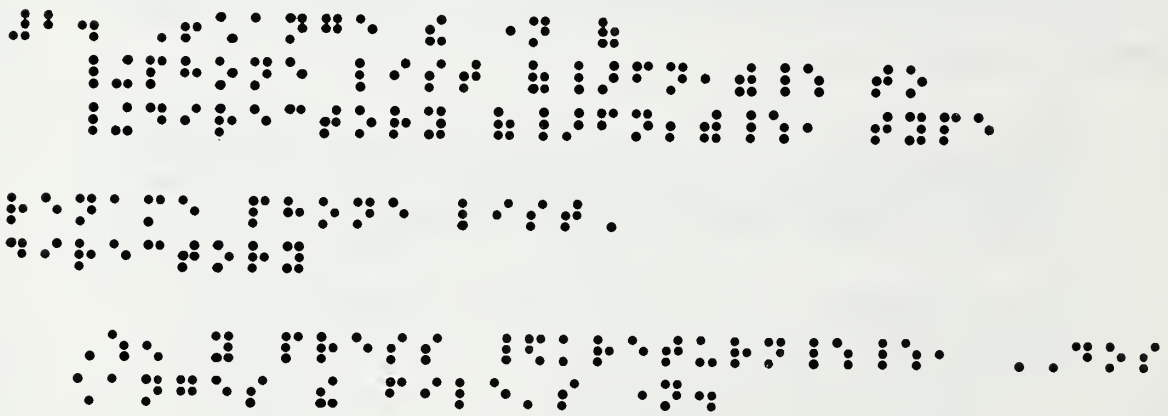


4. Stylized practice. Keys are used for material to be typed in by the user. A note on the Transcriber's Notes page at the beginning of the work should explain this usage. (Manuals for calculators would require the Nemeth Code key representations.) Computer notation is predominantly upper-case. Reverse the use of the Shift and Caps Lock indicators for lower-case letters.

2. To change the name of PHONE LIST (fn1) to DIRECTORY (fn2), type

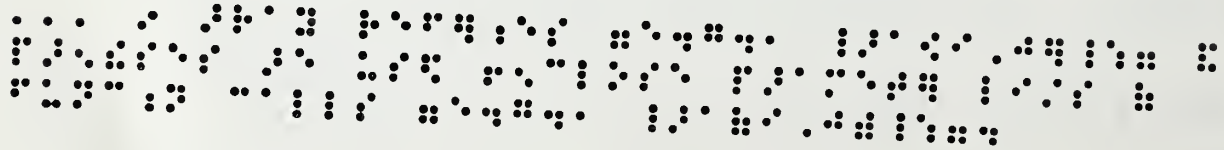
RENAME (SPACE) PHONE (SPACE) LIST.
DIRECTORY

When you press (RETURN), DOS changes the file's name.



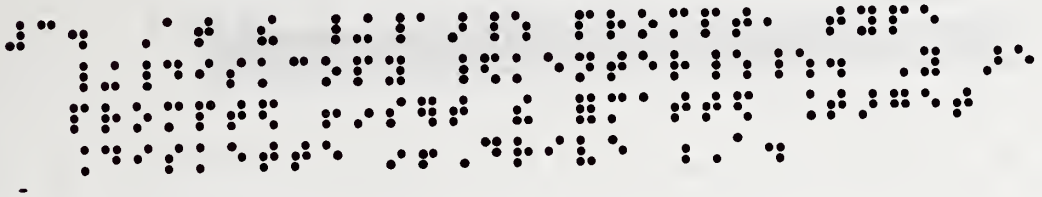
5. Italics are used for computer notation.

[Syntactic distinction is made—square brackets are used to enclose subscript lists in array references (e.g., $A[I,J]$) and parentheses are used to enclose parameter lists of function calls (e.g., $A(I,J)$).]



6. Boldface type is used for computer notation. [All letters in key representations are upper case. A transcriber's note should be included at the beginning of the book stating this fact.]

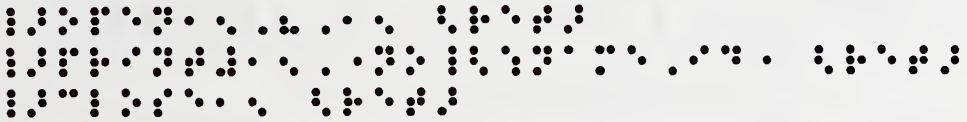
3. At the A> prompt, type Diskcopy ENTER. You are prompted to insert the formatted *target diskette* into Drive A.



7. ends when the closing brace of `main()` is reached.

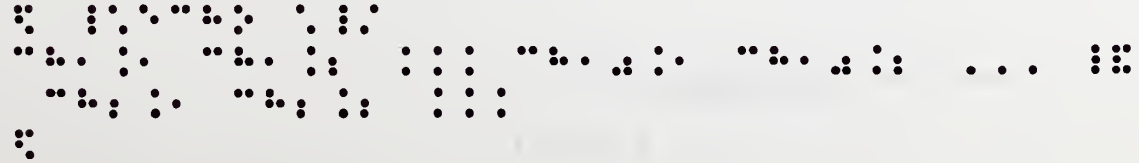
8. Dot matrix print. (Not all Dot Matrix is computer notation. See paragraph 5 in this section.) The Caps Release could come after the colon.

```
OPEN15,8,15 <ret>
PRINT#15,"N0:name,id" <ret>
CLOSE15 <ret>
```



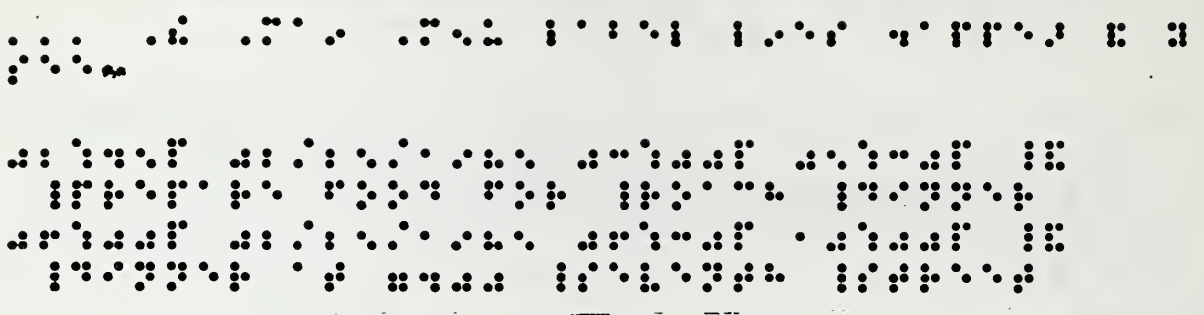
9. Two different type styles - block type is main computer notation. Slanted type is secondary notation.

```
$ echo *
ch1.1 ch1.2 ... ch10.1 ch10.2 ... ch2.1 ch2.2 ...
$
```



10. "Shaded" boxes [If another kind of print notation is used in this book, shaded boxes would require the Emphasis Indicator. Or, if all the material in the box can be done in literary, it could be done that way. See Section XXIV, "Screen Displays".]

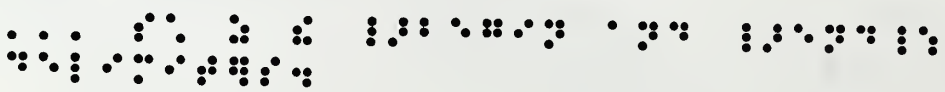
The Main Menu label lines disappear and you see:



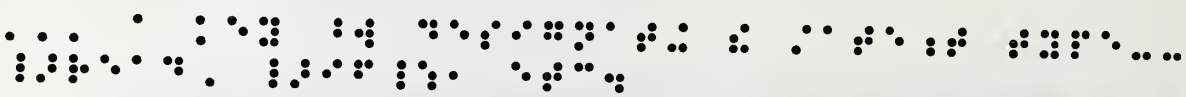
When no special type style or stylistic practice is designated as computer notation in the text - except possibly the use of upper-case letters in programs - the transcriber may have to exercise a great deal of judgment in determining material which should be transcribed as computer notation, especially when that notation is embedded. Here are some additional ways to help you distinguish computer notation.

1. Look for descriptive words in the surrounding text. (Underlining indicates the descriptive word.)

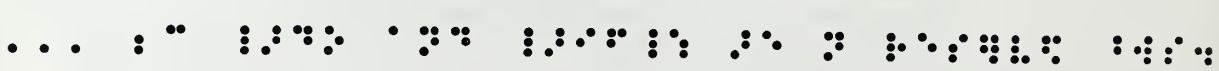
   a. ... such as the BEGIN and END delimiters.



   b. ... a key word designating the statement type READ, IF, etc.



   c. ... because DO and IF are not reserved words.



   d. Line 2 names the subprogram SUM ...

e. ... enter the <u>command</u> READ followed by ...

[illegible]

f. The <u>program</u> INTRINSIC is ...

[illegible]

2. If a word or term appears in the text and no description can be found in surrounding text, look for the word or term in a nearby program listing.

```
NEW
100  PRINT "WHITEDICE",
110  PRINT INT (6 * RND(1)) + 1
120  PRINT "REDDICE",
130  PRINT INT (6 * RND(1)) + 1
```

This program generates random integers from one to six for each die. To reroll the dice, reRUN the program. Can you write a program that uses these "dice" to play a game? Try it.

Try writing a one-line program that generates random numbers [from 1 through 50. From 0 through 25. Make up your own numbers. Remember to add (1) to the random number] if want to generate zeros.

(1) is included in the program.

[illegible]

3. Punctuation that may or may not be associated with computer notation can often be ascertained by looking for the notation in a program.

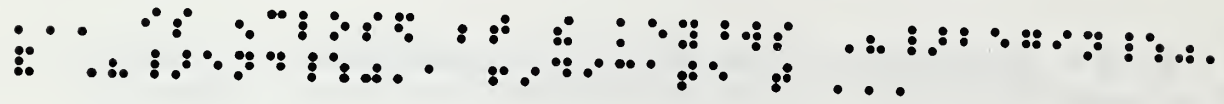
a. ... the program prompts for "SEQUENCE:", to which is added ...
("SEQUENCE:" appears in the program.)

[illegible]

b. PROGRAM heading. The text of the program is enclosed between the keywords 'BEGIN' and 'END', to indicate that this is the part that is actually executed. After the 'END' a full stop (.) is used rather than a semi-colon to indicate that this is really the end of the program. The full program now, therefore, will look something like this:

```
PROGRAM printit(input,output);

BEGIN
    <do something>
END.
```

(BEGIN and END do not have quotes in the program.)

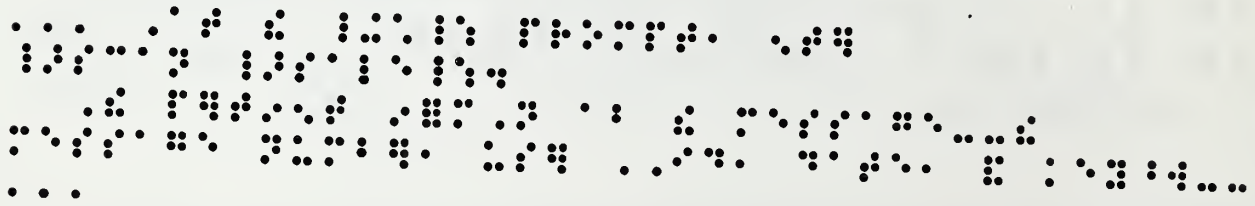


4. Some texts use the same type style for both computer notation and emphasis, i.e., upper case or italics. Distinguish through context, or other illustrations above, which is computer notation and which is not.

a.

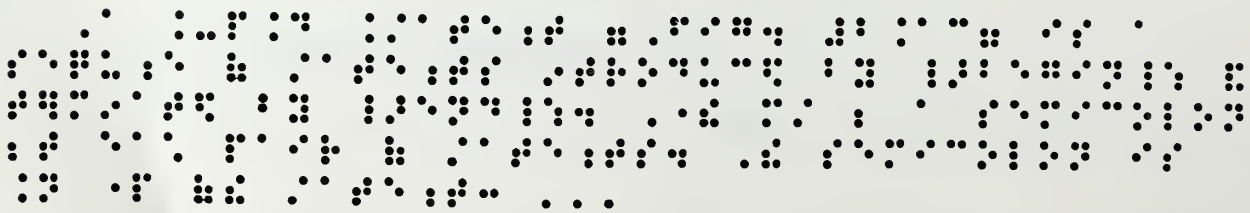
Ready? At the * prompt, enter SCAN SALE.

The pertinent information about the message--the message number, user ID, date, and keyword is summarized at the bottom.



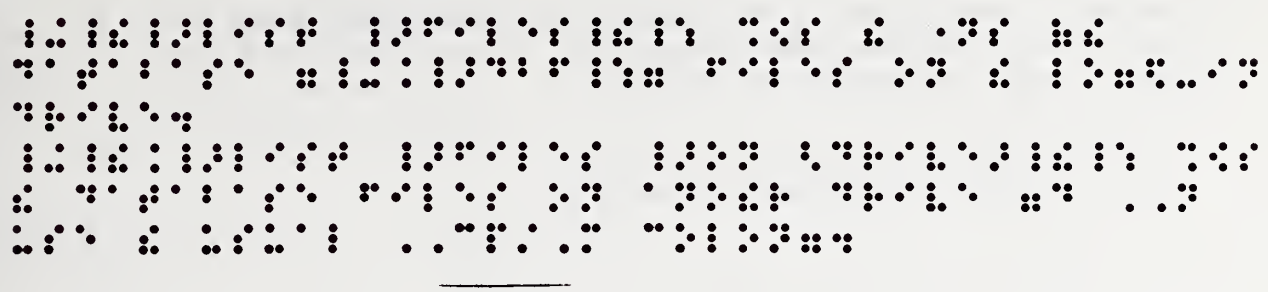
b. In this example, italics are used for both literary emphasis and computer notation:

A *compound statement* (Fig. 2.13) is a sequence of statements introduced by *BEGIN* and terminated by *END*. There must be a semicolon between each pair of statements. The semicolon is *not* part of the statement: it is a statement *separator*. Consequently, there is no semicolon between the last statement of a compound statement and *END*. Fig. 2.14 shows that a *statement* may be a *compound statement*.



c. In this example, upper-case is used for both computer notation and literary emphasis. The diamond shape is used in place of some literary closing symbol, such as quotes. This is explained in the text and, since there is no literary equivalent, the Transcriber's Option Symbol is used for the unusual shape.

◆LIST FILES◆ shows the names of the database (.DBF) files on the logged-in drive. ◆.LIST FILES ON <drive>◆ shows the database files on another drive (do NOT use the usual CP/M colon).



5. The following examples are from a text printed entirely in Dot Matrix print. Use illustrations above, and your common sense, to determine the terms that will be computer notation. (All dot matrix is not computer notation.)

a.

To format the disk, enter the commands below, where <ret> means to hit the RETURN key.

b. Load the program using the standard load command:
LOAD "TABICAT",8 <ret>

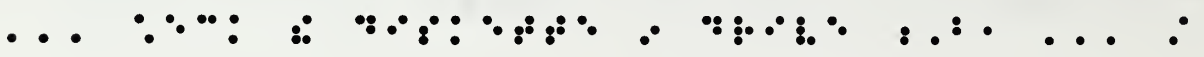
II. NOTATION IN THE PRIMARY CODE

The Computer Braille Code states (Section II-C) that "materials should be transcribed in the primary code unless there is good reason to enter Computer Braille Code." When the primary code is Nemeth, music braille, or Textbook Format Code, use the rules and symbols of that code for all material that is not specified as computer notation (by a special type style, etc.)

A. Textbook Format Code is the primary code most often used in transcribing computer-related materials. Using Textbook Format Code, transcribe notation as follows:

1. Single letters, upper or lower case.

a. ... check the diskette in drive B, ...



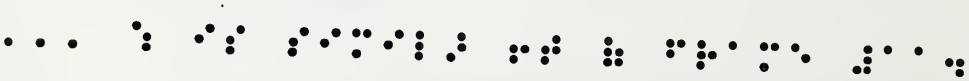
b. ... each k is a list containing ...

c. ... program control is transferred to c.

d. The statement numbers a, b, and c ...

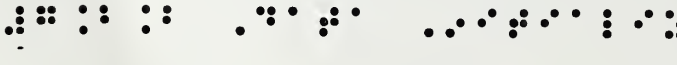
2. Numbers

a. ... which is similar to that of frame 11.



b. ... is a 1 to 4 digit sequence composed of ...

c. 7.2.2 Data Initialization Statement



3. Letters or numbers enclosed in parenthese, brackets, or quotes.

   Note: If you are using literary/textbook format for the primary code, remember that single letters enclosed in quotation marks are brailled as letters only.

   a. ... will now display an "S".

⠠⠠⠠ ⠠ ⠠⠠ ⠠⠠⠠⠠ ⠠⠠ ⠠⠠⠠

   b. (1) The only way to ...

⠠⠠⠠ ⠠⠠ ⠠⠠⠠ ⠠⠠⠠ ⠠⠠ ⠠⠠⠠

   c. ... use the (a), (b), (c) outline listings to ...

⠠⠠⠠ ⠠⠠⠠ ⠠ ⠠⠠⠠ ⠠⠠⠠ ⠠⠠⠠ ⠠⠠⠠⠠ ⠠⠠⠠⠠ ⠠⠠
⠠⠠⠠

4. Punctuation marks, whether they precede and/or follow primary notation or computer notation, except when the punctuation is part of the computer expression.

   a. The reason the variable was called "XMOVE" was ...

⠠⠠ ⠠⠠⠠⠠ ⠠ ⠠⠠⠠⠠ ⠠ ⠠⠠⠠⠠ ⠠⠠⠠⠠⠠⠠⠠⠠ ⠠
⠠⠠⠠

   b. A central processing unit ("CPU"; see Fig. 1) ...

⠠⠠⠠⠠⠠⠠⠠⠠

e. ... which is at coordinates (A, B); to the first ...

f. '1's. ['1' is a computer term]

5. Contractions and short forms (Exception: do not use (to), (into), or (by) before computer notation.)

a. ... is written (and "debugged") only once.

b. (More often than not they won't be.)

c. DO NOT use this string.

d. ... storage is limited to 128K bytes ...

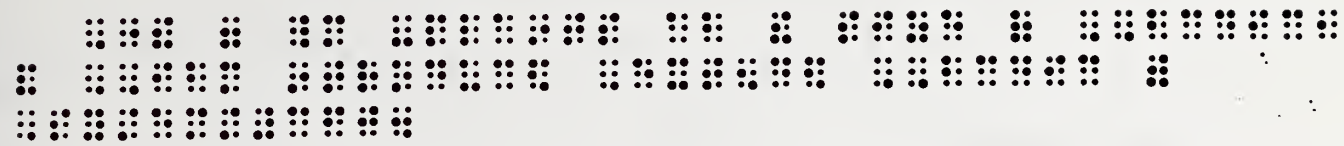
e. ... the arguments to SCANF must be ...

f. ... followed by y, and ...

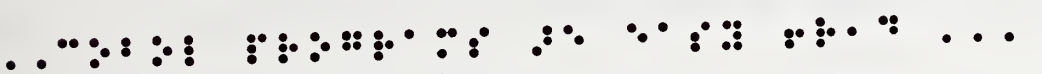
6. Acronyms and capitalized abbreviations

a.

There were many complaints about the size of BCDIC, and IBM introduced Extended BCDIC with System/360.



b. COBOL programs are easy to read ...



c. A listener must set NDAC low ...

d. FORTRAN stands for "FORmula TRANSlation";

e. Appendix A lists the ASCII character codes.

f. ... compute the COSine ...

g. ... a course at M.I.T.

h. Write a "BASIC-like" program ...

i. ... is the user ID number ...

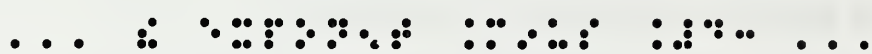
j. ... the DOS 3.3 SYSTEM MASTER disk ...

7. Abbreviations

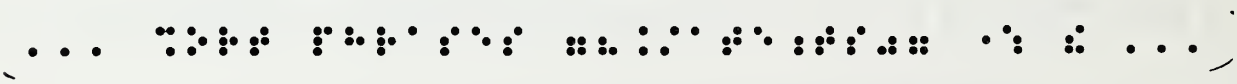
a. ... the readability of a variable name (e.g., hat_size).

8. Italicized letters, numbers, or words.

a. ... the exponent *minus 4:*



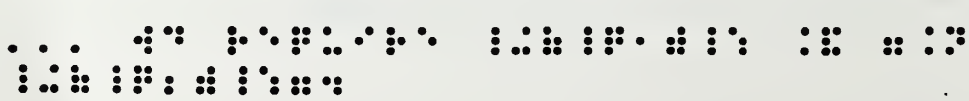
b. ... short phrases ("*statements*") where the ... [Quotes are not used in literary/textbook - the italics are omitted. But here the text stipulates that italicized words enclosed in quotes represent a particular meaning.]



c. ... a set of *nested Do loops*.

SUGGESTION: When an italicized phrase begins or ends with computer notation, or a 3-word phrase contains computer notation, use the italic signs before the italicized words only.

d. ... would require (Q1) *and (not* (Q2)).



9. Reference indicators (Also see Section VIII)

a.

Examples: (1) As an example, the MULTICS[2] timesharing system ...

... ...

[2]MULTICS = Multiplexed Information and Computer System

b. ... keeping recipients appraised of UNIVAC® Systems development.

... ...

®Registered Trademark of the Sperry Rand Corporation

10. Abbreviations of coinage, weights, and measures

a. 22 ft. 3 in.

b. 50¢

11. Roman Numerals

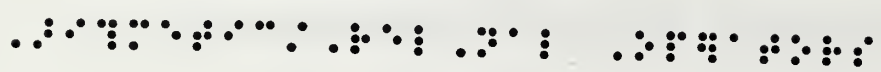
a. ... in Section I and Section V.

b. ... in the II and IV Sections.

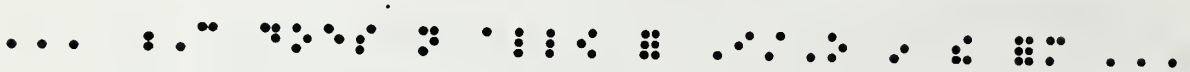
c. Use items i, iv, and viii.

12. Words or letters connected by a slash

a. Arithmetic/Relational Operators



b. ... C does not allow for I/O in the form ...

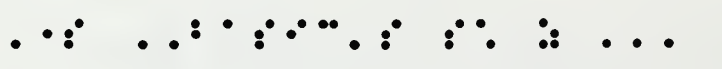


c. and/or

---

13. Letters, numbers, acronyms, etc. followed by the letter s or apostrophe s

a. Some BASIC's such as ...



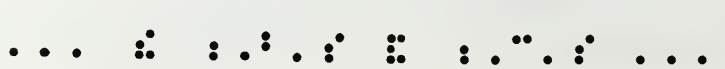
b. ... C's standard library ...

c. Scan the 0's and the 1's.

d. ... in the mid-1960s. [NOTE: Use the English Braille code rule for inserting an apostrophe.]

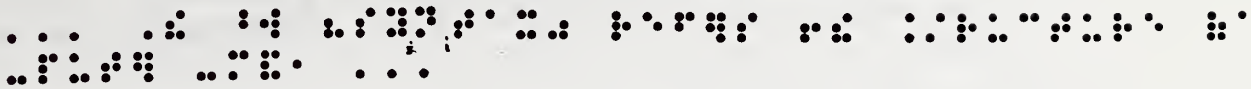
e. IBM'S programs ...

f. ... the Bs and Cs ...



14. Boldface, italicized, capitalized or underlined type used for literary emphasis

a. There are two terms you'll need to know when reading this manual. The word "syntax" refers to the structure of a computer command, the order and correct form of the command's various parts. The word "parse" refers to the way in which the computer attempts to interpret what you type, picking out the various parts of the computer commands in order to execute them. For



b. [Dot matrix type is not always computer notation.]

The SPACE bar (Figure 2-2) at the bottom of the keyboard is another key whose role you must understand. Pressing the SPACE bar generates a space character. That's right, a space is a character—and a very important one at that.

c.

The next program reads four fields of the sequential-access text file FOUR.FRUITS into successive elements of an **array**. The DIM statement in line 15 sets aside space for an array of the specified number of **elements**.

d.

The ESC key is the last and most important of the control keys. This key is often used to put the computer into **escape mode** or to begin an **escape sequence.** Examples of escape sequences occur in many programs written for Apple II systems.

SUGGESTION: When both boldface and italics (or any of the above type styles) are used for emphasis in literary passages, use the italic sign for both unless there is a very good reason to show a distinction (e.g., if the text states a particular use for each.) If there is a need to distinguish one from the other, use the boldface type indicators of the Textbook Code. These symbols must be included on the "Special Symbols" page in each volume. (Also see Section XI - note.)

15. Transcriber's Notes. Use the symbols and rules of Textbook Format Code.

16. When mathematical expressions cannot be transcribed in literary braille, use computer notation. (See Section IX and Section XXIII.)

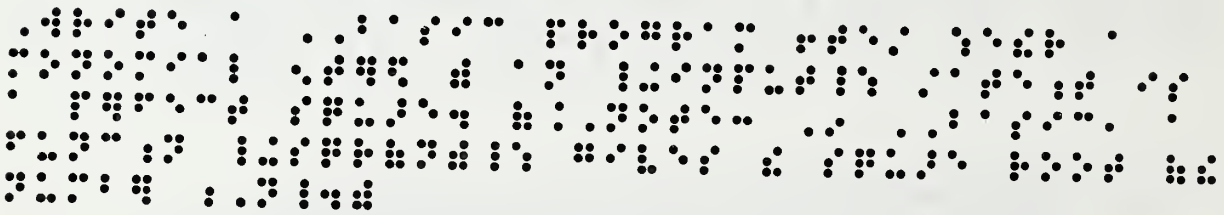
17. When mathematical expressions cannot be transcribed in computer code, use Nemeth Code Indicators and Nemeth Code notation. (See Section VIII)

B. When Nemeth Code is the primary code - e.g., an algebra book with sections having to do with the use of computers or computer programming - use the Guidelines to determine the material that is computer notation. This material will virtually always be printed in a different type style from that used in the mathematical portions of the book. Use the rules and symbols of the Computer Braille Code for the computer notation only. Use the rules of the Nemeth Code for all other material, even if it is labeled "Computer" in some way (numbers, letters, mathematical expressions, etc.). Remember, Nemeth is the primary code.

1. [Block letters are Computer notation; italicized letters are Nemeth Code.]

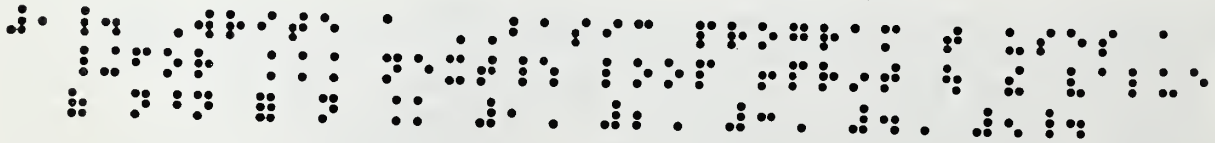
Write a BASIC program to test whether a monomial entered with an INPUT statement is a perfect square (*Note:* In BASIC, the function SQR(N) gives the square root of the number *N*.) Use the program to test whether each of the following monomials is a perfect square.

1. 36
2. $4x^2$
3. $16y$
4. m^2n^4
5. $2xy^6$
6. $25w^9$



2.

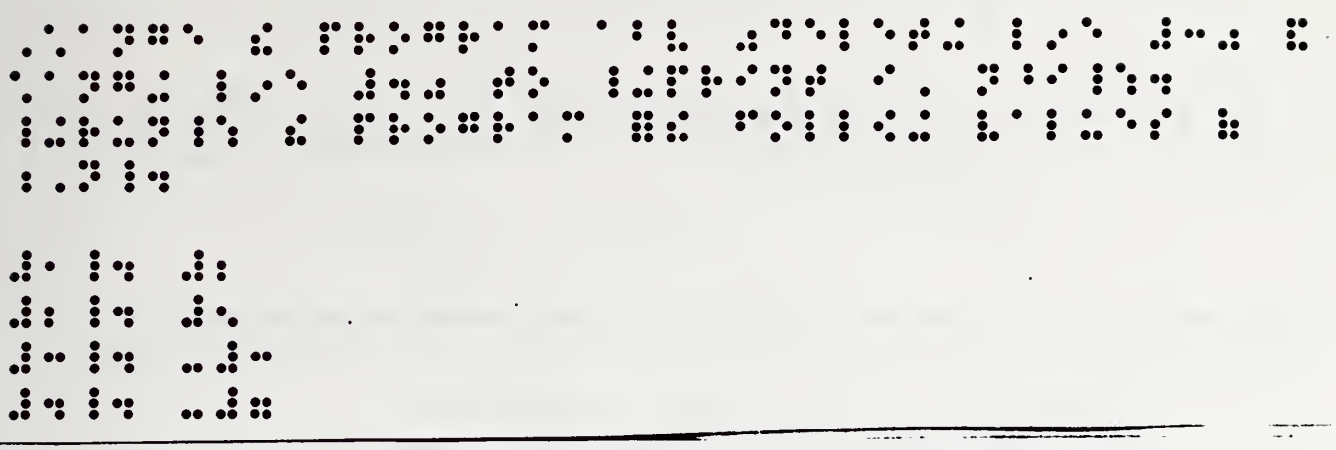
1. Write a BASIC program that uses a FOR . . . NEXT loop to print out the value of n^n for $n = 1, 2, 3, 4, 5$.



3.

Change the program above by deleting line 30 and changing line 40 to PRINT I, N↑I. RUN the program for the following values of N.

1. 2 2. 5 3. −3 4. −7



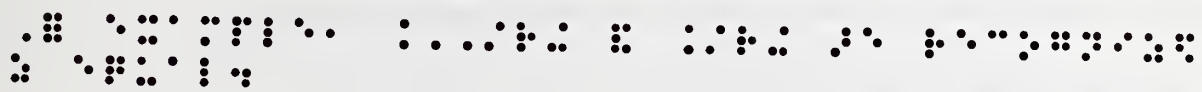
4.

A computer will express a rational number, such as $\frac{4}{7}$, as a decimal by having it PRINT 4/7. The result will display the fixed number of decimal places

PRIMARY CODE - ADDITIONAL EXAMPLES

1. [Italicized words are literary emphasis to point out that there is no difference between upper and lower case notation in the programs of this book.]

   [For example, *STRING* and *string* are recognized as equal.]

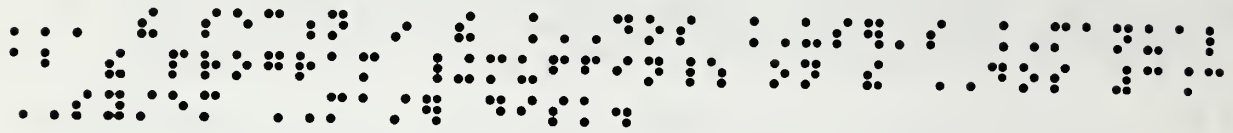


2. [Although there are no spaces in print between the numbers, it is obvious that this is not computer notation.]

   ... is a power of 2 (e.g., 2,4,8,16, etc.).

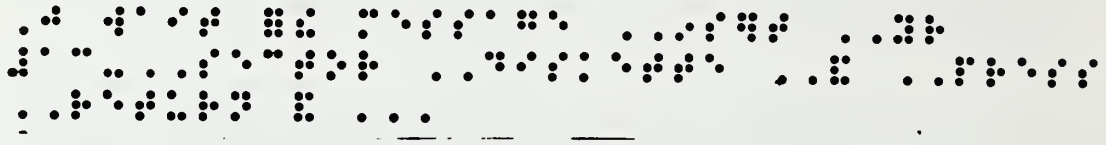
3. [Boldface and regular upper-case for primary code. Dot matrix for computer notation.]

After you're a bit more comfortable with your computer, you might want to read the section in the **DOS User's Manual** about the program MUFFIN on the DOS 3.3 SYSTEM MASTER disk.



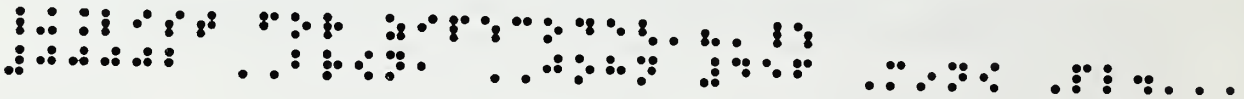
4. [Dot matrix, but can be done in literary. Screen messages are usually in the primary code. They are almost always literary text.] (See Section XXIV.)

disk in drive 1 and start or restart the system. You don't have to type anything. Just wait for the message `INSERT YOUR 13-SECTOR DISKETTE AND PRESS RETURN` and be sure to put the 13-sector disk in drive 1.



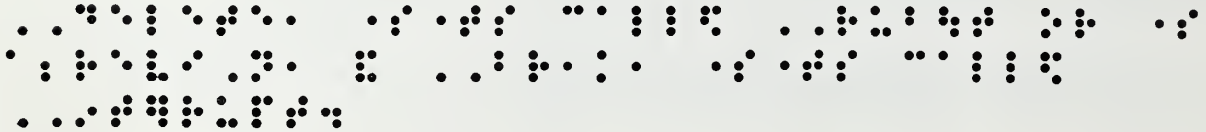
5. [When most of a printout can be done in literary]

```
. list for zip:code < '8'
00002  BROWN, JOHN       456 Minnow Pl.     Burlington   MA 01730
00004  DESTRY, RALPH     234 Mahogany St.   Deerfield    FL 33441
00006  FORMAN, ED        456 Boston St.     Dallas       TX 75220
```



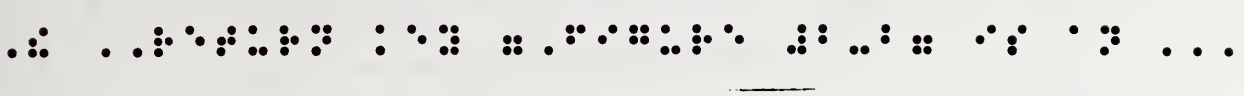
6. [Upper case for literary emphasis. Key names can be literary.]

Two other keys have special meaning: DELETE, sometimes called RUBOUT or some abbreviation, and BREAK, sometimes called INTERRUPT. On most UNIX systems, the DELETE key stops a program immediately, without waiting



7. [Literary numbers and key name]

The RETURN key (Figure 2-2) is an important key on the keyboard. It serves two purposes:



8. [Programs are in italics - but so are literary emphasis.]

```
WHILE boolean expression DO
   BEGIN
      statement1;
      statement2;
      ...
      statementk
   END
```

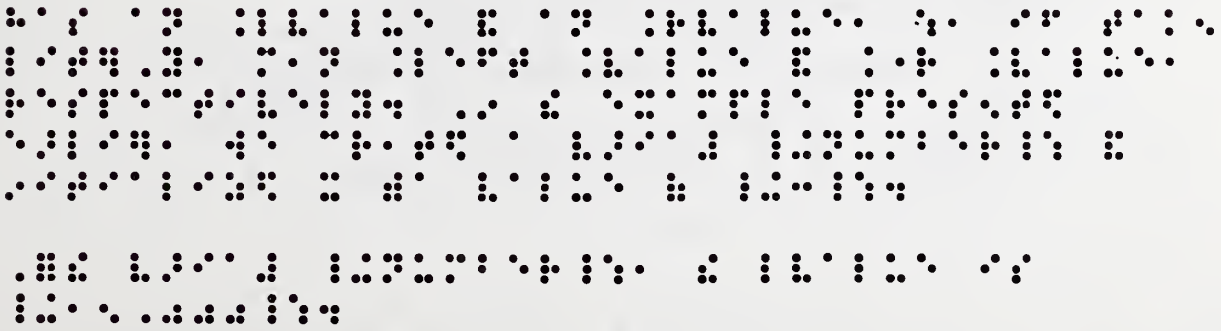
Although statements and compound statements are each defined partly in terms of the other, the definitions are not circular, as you will discover by examining the syntax diagrams. They are *recursive* definitions. The fact that PASCAL is a very rich language despite its relatively simple syntax is largely due to the careful use of recursion in its design.

9. ["Words" are not strictly English, but don't need to be computer notation, which is in boldface type.]

Every data item in a language has an *lvalue* and an *rvalue*, which, if taken literally, mean *left value* and *right value*, respectively. In the example presented earlier, we created a variable **number** and initialized it with a value of 3. We also saw that the compiler placed **number** in memory starting at address 15,000.

The *lvalue* (left value) of a data item is the address in memory *where* that item is located. For the variable **number**, the lvalue is **15,000**.

The *rvalue* (right value) of a data item represents *what* is stored at the data item's lvalue. In other words, the rvalue is what has been assigned to the data item. We can now visualize these two concepts as

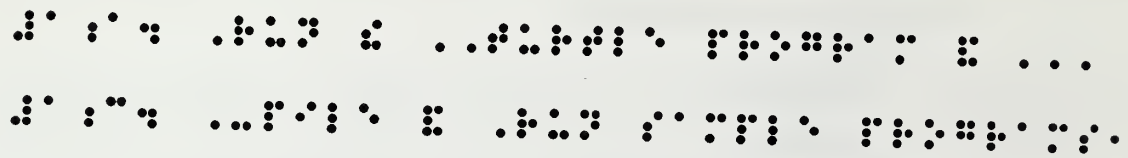


10. [Numbers in primary code. If all program names are words, use literary.]

1a. Run the TURTLE program and use it to draw pictures on the computer's display screen.

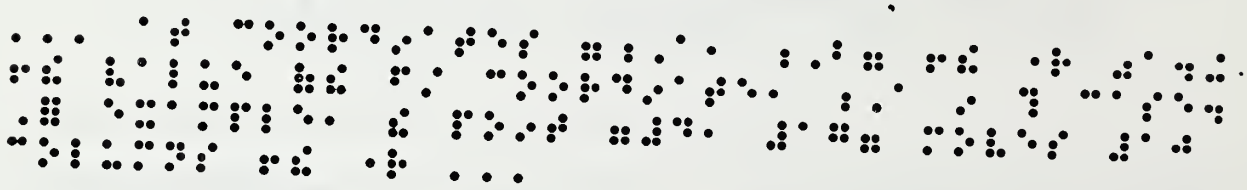
1b. Learn to use the computer's Editor to read and modify sample programs.

1c. Compile and Run sample programs, returning to step 1b several times until you understand what they do, and how to change them to obtain different results.



11. [Not computer notation. Insert spaces in order to do literary.]

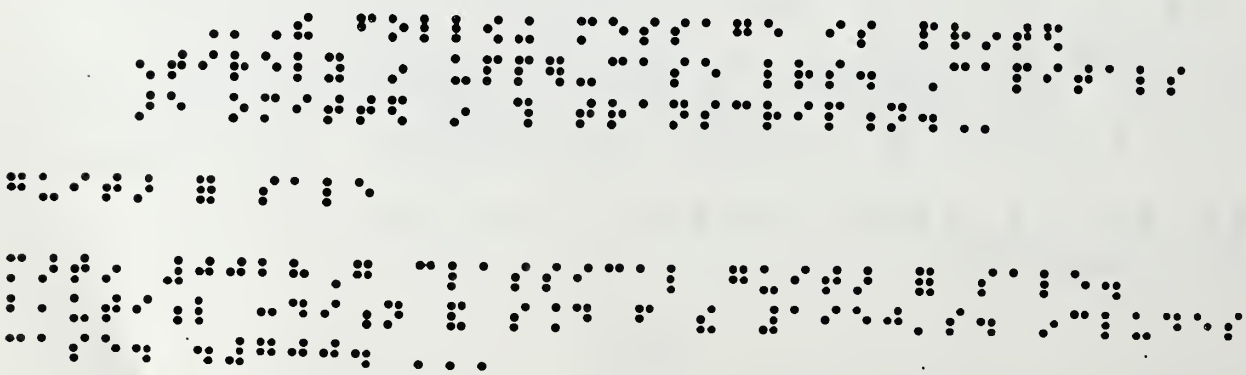
The key to doing this begins with the observation that you move a point which is at coordinates (A,8) to the right by adding to the value of the first coordinate, A in this case. For example, the point (4,17) moves 10 columns to the right.



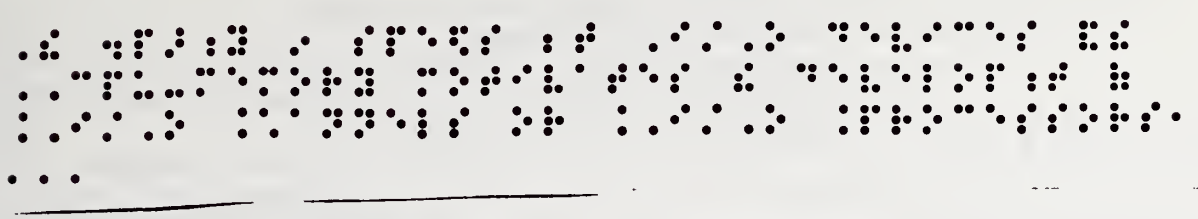
12. [Use literary for screen displays (see Section XXIV) using computer notation where necessary. A transcriber's note should be included explaining that all letters are upper case.]

You should then see that message displayed, as we did:

```
GUITAR FOR SALE

MARTIN 0028G CLASSICAL GUITAR FOR
SALE. BEAUTIFUL CONDITION AND SOUND
FROM THE 1950S. INCLUDES CASE.
$7000 ALSO FAIRBANKS & COLE 1890S
BANJO. FAIR CONDITION $300. CONTACT
FRED 619-555-9993 OR 70000,1101.
```

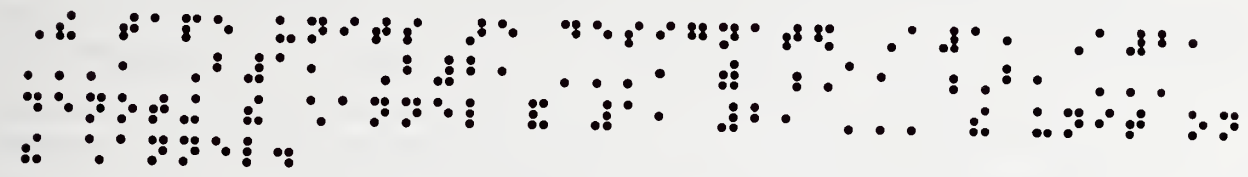


13. The disparity in speeds between I/O devices and the CPU-memory motivates the development of *I/O channels* or *I/O processors*, ...



14. [Literary, with spaces inserted]

The tape units are designated A1,A2,...,B1,B2,..., with A,B,... denoting the channel and 1,2,... the unit on the channel.

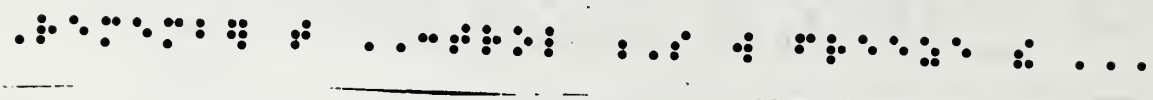


15. syntax diagrams. Inside the boxes of a syntax diagram, we will omit the broken brackets, but will still use lower-case characters for items that are defined by other syntax diagrams. [A word appearing in UPPER CASE letters in a syntax diagram is a "*Reserved Word*" having special significance to the compiler.] A reserved word should appear in a program with the same

16. appear in that line. Remember that all of the alphabetic characters [are greater than or equal to 'A' AND less than or equal to 'Z', OR they are greater than or equal to 'a' AND less] than or equal to 'z'. Test your program using any five lines of text from this book which contain

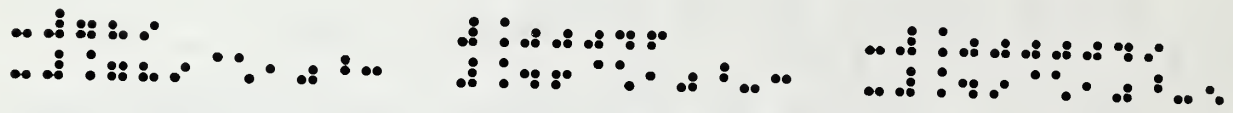
17. *Conditional Statements.* An ALGOL-like IF ... THEN ... ELSE ... ; conditional statement is used, with the ELSE ... part

18. [Remember that CONTROL S will freeze the screen if you can't keep up with the scroll, and CONTROL Q will start it again.



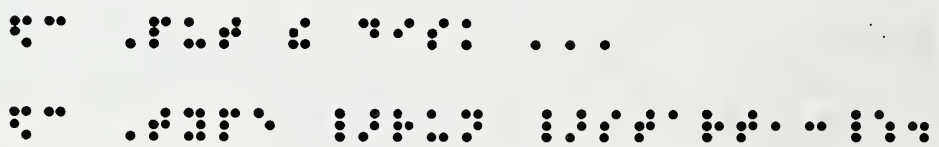
19. [The primary code is Nemeth Code.]

-789	.00046	-.0000049
$-.789 \times 10^{3}$	$.46 \times 10^{-3}$	$-.49 \times 10^{-5}$



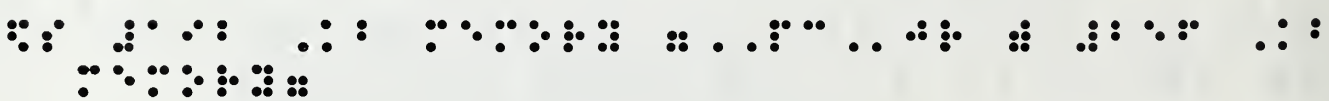
20. If you plan to use a 13-sector disk only once in awhile or if it is copy-protected, do a two-disk startup:

- Put the disk labeled DOS 3.3 SYSTEM MASTER in drive 1, close the drive door, and start or restart the system.
- Type RUN START13.



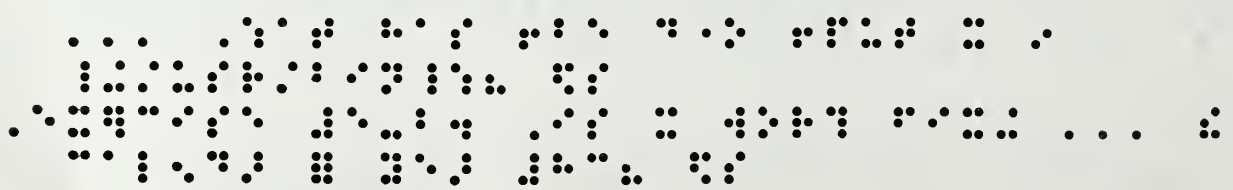
21. *Executive Filer* runs on the IBM Personal Computer with:

- 192 Kb memory (PCjr with 256 Kb memory)
- One diskette drive (two disk drives are strongly recommended)



22. **Exercise 5-1.** If users prefer your version of `cal`, how do you make it globally accessible? What has to be done to put it in `/usr/bin`? □

Exercise 5-2. Is it worth fixing `cal` so `cal 83` prints the calendar for 1983? If so, how would you print the calendar for year 83? □



III INDICATORS

When computer notation appears in textual passages (embedded) it must always be introduced by one of 5 computer code indicators.

The indicators are
- Caps Lock Indicator
- Begin Emphasis Indicator
- Begin Shape Indicator
- Begin Nemeth Code Indicator
- Begin Computer Braille Code Indicator

Each beginning indicator initiates a specific kind of computer notation, and also initiates general computer notation as well as the particular construction for which it is intended. Consequently, computer code symbols are used freely with these indicators in accordance with the rules of the Computer Braille Code.

Each of the first four indicators listed above may be used within embedded notation, or in displayed notation.

The Computer Braille Code Indicator is used only to initiate computer notation in embedded notation. It is never used in displayed notation that is set off from the body of the text by blank lines, and it is never used in embedded notation that has begun with another indicator. (See Section IX.)

Other indicators used within computer notation are:
- Shift/Lower Cell Sign
- Caps Release
- End Emphasis
- Continuation
- Countable Spaces
- Half-Line Shift Up
- Half-Line Shift Down
- Transcriber's Primary Option Symbol
- Transcriber's Secondary Option Symbol

Each of these indicators will be discussed in the sections which follow.

Material that is within the effect of any indicator must be transcribed character by character using Computer Braille Code letters, numbers, and symbols. Numbers are transcribed in the lower two-thirds of the cell. Capital signs, letter signs, and number signs are not used.

Computer notation in textual passages (embedded) must always be terminated with the Termination Indicator to signal a return to the primary code. (See Section XIII)

In the sections which follow, each indicator is described and the rules for its use are illustrated with examples.

IV SHIFT INDICATOR
(Section IV - CBC)

Unless otherwise indicated, the Computer Braille Code assumes that letters and words contained in computer notation are lower case. Therefore, indicators have been assigned to indicate the occurrence of upper-case letters in computer notation.

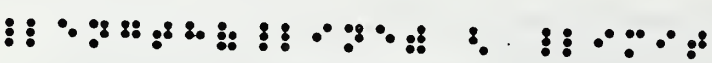
The Shift Indicator is used to indicate a single upper-case letter. A single upper-case letter is defined as one that is preceded and followed by spaces and/or lower-case letters. The spaces and lower-case letters may have other computer symbols included before or after them. The Shift Indicator affects only the letter it precedes. (An upper-case letter that is preceded or followed by other upper-case letters, without an intervening space, whether or not other symbols intervene, is not a single letter.) (See Caps Lock Indicator, Section V.)

1. The Shift Indicator must precede a single upper-case letter when it occurs in displayed computer notation.

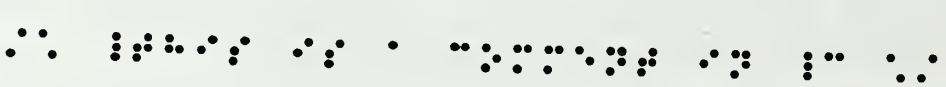
   a. A> DIR

   b. A: FORMAT

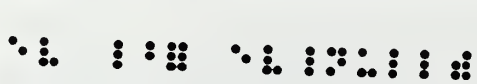
   c. Length(Line) < Limit

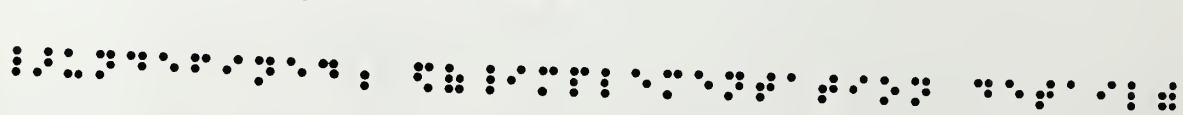
   d. /* This is a comment in C */

   e. (ev ~= evNull)

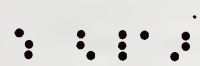
   f. UNDEFINED; $(Implementation detail)

g. VFUN PSTmsgToVec(ipcMsg

h. Union Order Package =

i. : <A>



j. S ⊟ R =

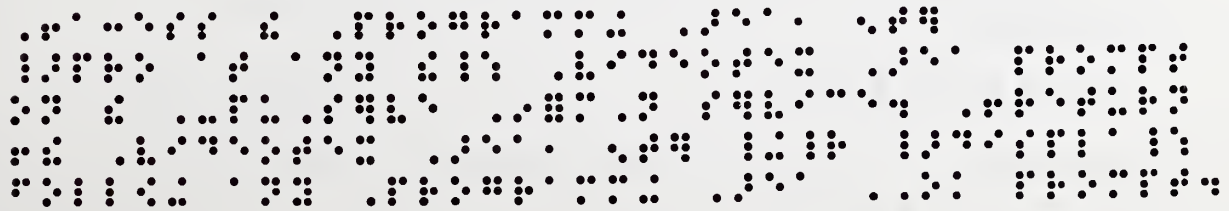
2. When a single upper-case letter occurs in embedded computer notation, the Shift Indicator is used as above. In addition, if the upper-case letter is the first character of the notation, it must be preceded by the Computer Braille Code Indicator as well as the Shift Indicator. (See Computer Braille Code Indicator, Section IX.)

a.

This time, type Dallas (ENTER) for the search string and **Fort Worth** (ENTER) for the replacement string.

b.

To access the Programming Area, enter **PRO** at any ! Videotex Area prompt on the CompuServe Information service. To return to the Videotex Area, enter **R DISPLA** following any Programming Area OK prompt.



c.

... to say X = 30*y; ...

V CAPS LOCK AND CAPS RELEASE INDICATORS (Section IV-B, CBC)

The Caps Lock Indicator is used to indicate the occurrence of a string of two or more upper-case letters, whether consecutively or separated by any computer symbols other than spaces.

1. When such a string occurs as the first characters of computer notation, whether displayed or embedded, the Capts Lock Indicator must be the first symbol of the notation. (But see #6 in this section.)

   a. FORMAT.COM:

   b. PGM=LIST

   c. A(1)+A(2)+A(3)

   d. PRINT3

   e. AVG:=SUM/N

2. When such a string occurs within notation begun with symbols or letters other than upper-case letters, the Caps Lock Indicator must immediately precede the upper-case string. In displayed items:

   a. <RET>

   b. //LISP

   c. ev:BOOLEAN

   d. ε CompareAB

e. 1.5*(Z(I+1)-Z(I))

In embedded items:

The same rules apply, but notation will begin with another indicator. The use of the Caps Lock Indicator in this type of notation will be covered with each indicator. Each of the examples in 2. above would begin with the Computer Braille Code Indicator.

3. When such a string is followed, without a space, by lower-case letters, whether or not other symbols intervene, the string of upper-case letters must be terminated by the Caps Release Indicator. (also See Section XXIII)

a. COM<cr>

b. PSTmsgTo

c. (=MAX-1;i++)

d. [:d]⠸TO⠸filename

e. SEQUENCEd

4. A space terminates the Caps Lock Indicator. Consequently, when notation begun with the Caps Lock Indicator is followed by a space and then another string of upper-case letters, the Caps Lock Indicator must be repeated.

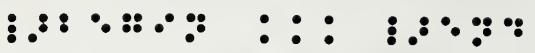
a. DIR *.COM

b. EXEC PGM=LISP

c. //STEPLIB DD DSNAME=LISP

d. VFUN PSTmsg

e. BEGIN ... END



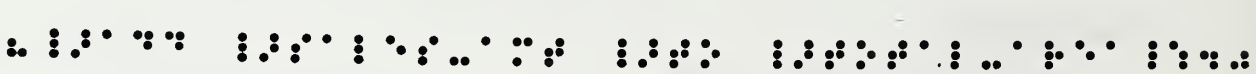
f. WHILE1, WHILE2, or PLOTNAME

5. A Termination Indicator must be used to signal the return to the primary code when notation begun with the Caps Lock Indicator occurs in textual passages. (See Section XII)

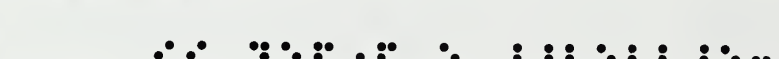
a. To OVERLAY one character on top of ...

b. ... all files named ACCOUNTS, regardless ...

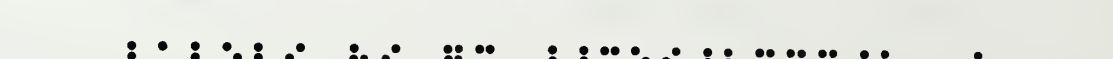
c. "ADD SALES-AMT TO TOTAL-AREA."



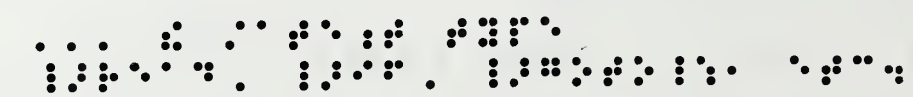
d. ... is defined as BELL.



e. ... labels of the form COSxxx where ...



f. ... the statement type READ, IF, GOTO, etc.



6. When computer notation throughout a text is exclusively upper-case, the transcriber may elect to use single-cell letters to indicate upper case. (See Section IV.A. in CBC.)

Many computer languages use upper-case letters exclusively. Even in some languages where lower-case letters are used, or are permissible, the author of a text may use only upper-case letters in order to show programs, etc., in a more highlighted manner. If, when setting up the text, you see that all, or virtually all, lines in programs are printed in upper-case, and that the text is also full of upper-case words and letter strings, use this method to transcribe the text. It eliminates the necessity of using the Caps Lock Indicator before all of these items.

When the upper-case letters or words are embedded, the Computer Braille Code Indicator must be used preceding the notation, just as it would be used if the notation were in lower-case letters, and the Termination Indicator must be used to signal a return to the primary code.

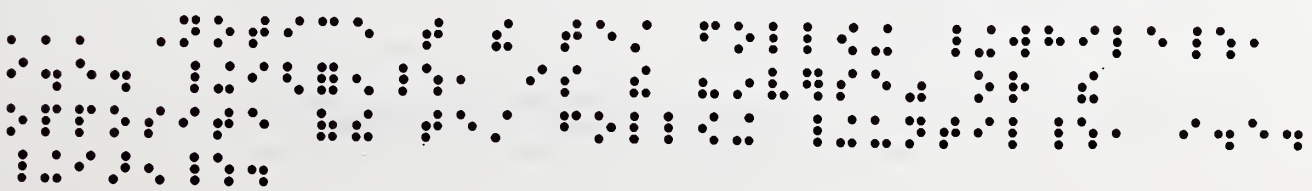
(See Section IX)

A transcriber's note must be included at the beginning of the book to explain this usage.

a. Write a BASIC program to test whether a monomial entered with an INPUT statement is a perfect square. [*Note:* In BASIC, the function SQR(N) gives the square root of the number *N*.] Use the program to test whether each of the following monomials is a perfect square.

b.
```
PROGRAM WHILEDEMO;
VAR I : INTEGER;
BEGIN
  I:=1;
  WHILE I<=5 DO
    BEGIN
      WRITELN(I);
      I:=I+1;
    END;
END.
```

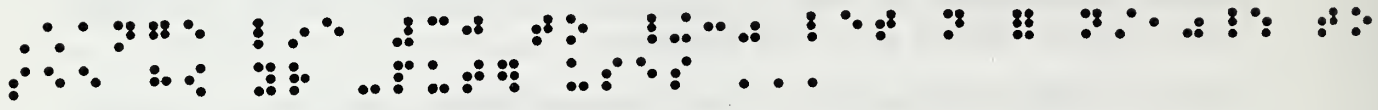
Each program should display a column of the five integers from 1 through 5. [Notice that the test following WHILE, i.e. I<=5, is the "*inverse*" or the opposite of the test following UNTIL, i.e. I>5.]



c.

```
10  LET N=1
20  FOR I=1 TO 39
30  LET N=N*10
40  PRINT N,
50  NEXT I
60  END
```

[Change line 30 to 30 LET N = N/10 to see how your computer uses scientific notation for small numbers.]

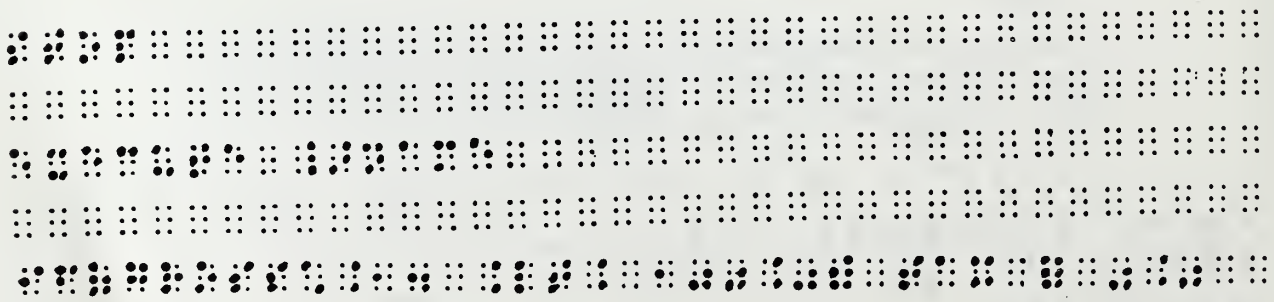


A text that is "predominantly" upper-case will have some instances where lower-case letters are used for a purpose other than program line notation and these letters may occur in either or both displayed and embedded material. For example, an occasional author's comment may appear in a program or program line, or lower-case letters may be used for a component of a syntax statement.

In this case, the Caps Lock indicators and rules can be used to transcribe the lower-case letters. The Shift Indicator would also be used where necessary for a single lower-case letter. A transcriber's note should be included at the beginning of the work to explain this procedure. (Also see the suggestion in Section XXIII for eliminating indicators in this case.)

1. [A rare occurrence of lower-case]

```
EXIT

NEXT

[ STOP

EXECUTE name

IF(GROSS*.14 .LT. 100.0) TAX = 0.0]
IF(FOUND) RETURN
```



Examples 2 and 3 illustrate lower-case in syntax diagrams only.

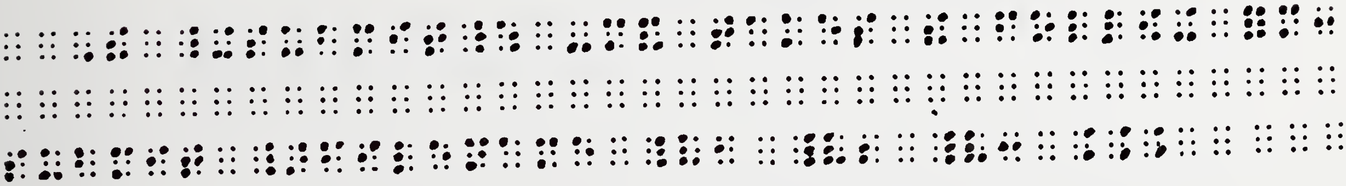
2\. The command has this syntax:

[VERIFY fn [,Sn] [,Dn] [,Vn]]

3\. The SUBMIT program first looks for SAMPLE.SUB, and then starts executing the commands. To execute DIR, it plugs the value 'B' into $1 and 'FILE1.TXT' into $2, and displays the filename in the directory for drive B. Then it executes PIP to copy B:FILE1.TXT onto drive A, using the same name.

[The SUBMIT command takes the following form:

SUBMIT filename v1 v2 v3 . . .]



4\. This program poses a problem. Although there are many lower-case notations, they are virtually all "comments" which are always started with a single quote symbol. There are also quite a few lower-case notations that are a part of the program and which are always enclosed in double quote symbols. The procedure which would be easiest for the reader would seem to be as follows:

Transcribe all upper-case notation in single cell letters.

Place the "comments" in cell 5 following the program line to which they refer. Omit the single quotes and use literary braille for these.

Reverse the Caps Lock rules for lower-case that is included in program lines.

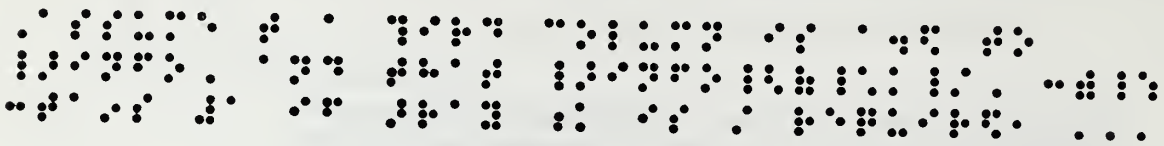
Transcriber's notes at the beginning of the work should explain these procedures.

```
4600    GOSUB 4720                 'Print check
4610    PUT #1.RN
4620 RETURN
4630
4640 'Print alignment mask
4650
4660 LSET EMPNAME$ = "THIS CHECK IS VOID": NETPAY# = 123.45
4670 GOSUB 4720                    'Print a check
4680 RETURN
4690
4700 'Print a check
4710
4720 LPRINT:LPRINT:LPRINT TAB(60);DATE$
4730 LPRINT:LPRINT "Pay to the order of: ";EMPNAME$:
4740 LPRINT TAB(60) USING "$$###.##";NETPAY#:LPRINT:LPRINT "The sum: ";
```

CAPS LOCK INDICATOR
Additional Examples

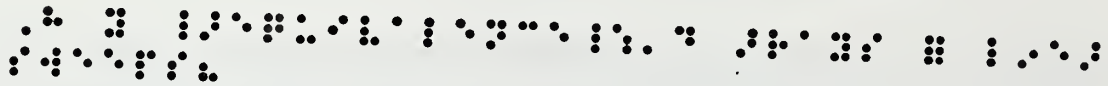
1.

Assume that a third column is added to INFO, and that INFO(k, 3) contains 1 if array k is still required, ...



2.

Have you EQUIVALENCE'd arrays for linear sweeps?



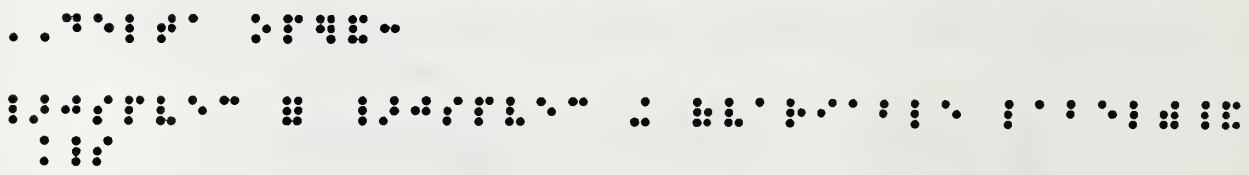
3.

```
... AAA.BBB.CCC, ...
```

4.

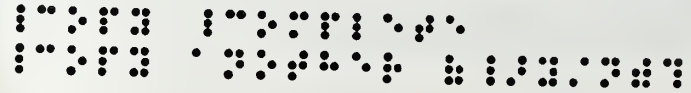
DELTA operand:

```
JSPVEC = JSPVEC + (variable label).S
```



5. [Section XXIV]

```
Copy Complete
Copy another (Y/N)?
```



6.

The number, n, is the slot of the device from which to get information. The number can be from 0 to 7. When you use the IN# command, DOS will get information, or *read*, from the device connected through

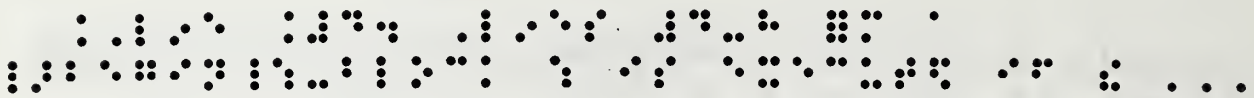
7.

```
while ((type = getop(s, MAXOP)) != EOF)
    switch (type) {

    case NUMBER:
        push(atof(s));
        break;
```

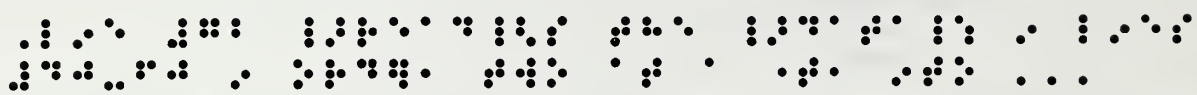
8. [Entire program upper case]

```
=AJ830
=BETA
=CRT
=DIABLO
=FRIDEN
=GE 300
=HAZELTINE 2000
=LOGABOX
=MEMOREX
=NCR 260
=QUME
```

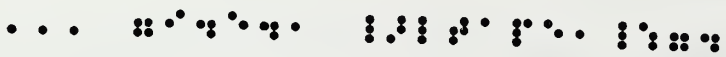
9. *Line 4.* Lines 4—8 form a BEGIN-block which is executed if the



10. Line 72 READs the DATA in lines 40–60 in order, two at a time, into

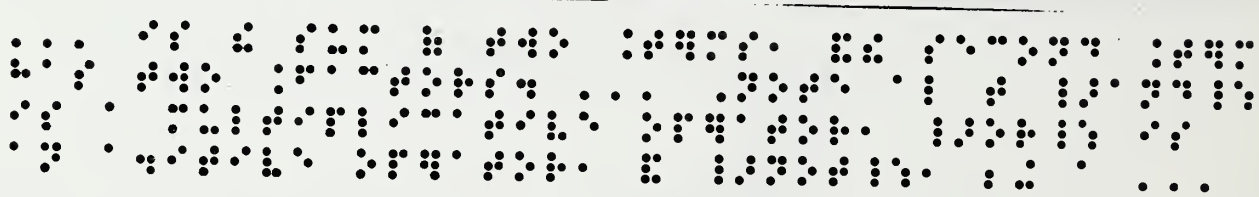


11. ... (i.e., LTAPE1).



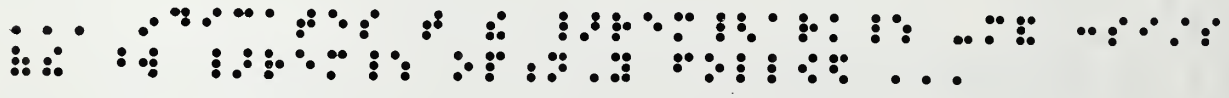
12. [Both literary and computer notation are in italics.]

[is the sum of two *terms*, and the second *term* has two *factors*.] Multiplication of factors is performed before terms are added. [Note also that *AND* is a multiplicative operator, *OR* is an additive operator, and *NOT*, being a part of *factor* syntax, takes precedence over either.



13. The words would come from locations X,X+1,...,X+19,Y,Y+1,...,Y+4.

14. [indicates that the REMark command consists of the word REM optionally followed] by one or more characters.



15.

A2=A1+B2, X2=(Q+B2)

VI EMPHASIS INDICATORS
(Section X - CBC)

Quite often in computer-related materials more than one type style is used to distinguish between different items of computer notation. (See page 18, Paragraph A of the CBC.) This usage is almost always either explained in the text or is obvious to the eye. The type style that is used for "programs", or that occurs most frequently in programs, is usually the primary CBC notation. The secondary type style should be indicated by the emphasis indicators. In all of the examples in this section, the Emphasis Indicator is used with a type style that has been determined to be the secondary computer notation.

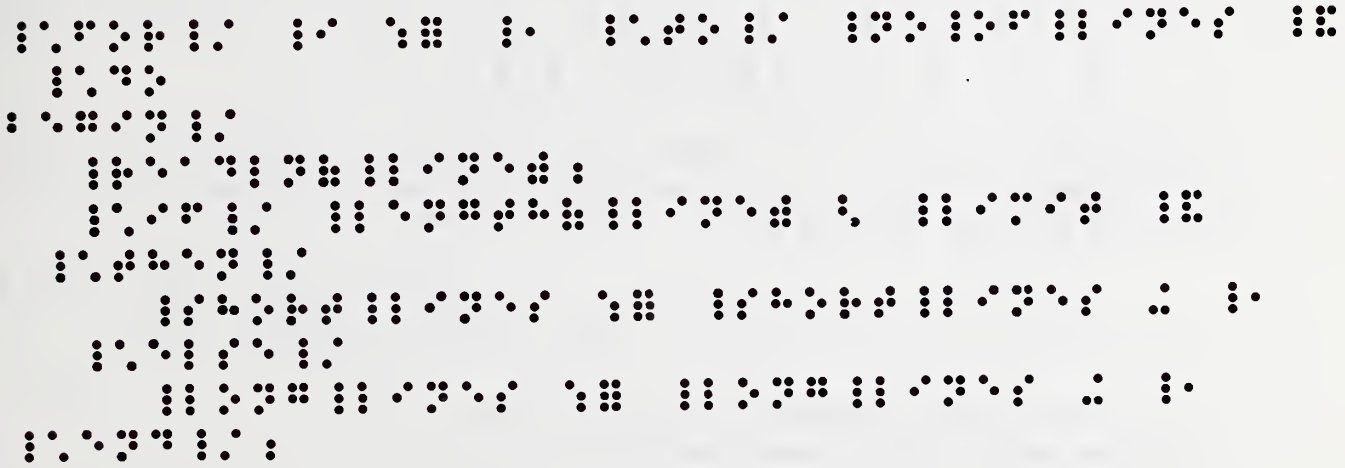
It should be pointed out that the print convention indicated by the Emphasis Indicator is not generally computer notation that is meant to be "emphasized" in the text. In fact, the Emphasis Indicator always indicates a secondary type style when more than one type style is used for computer notation. Consequently, if, as in Example 2 below, computer notation throughout the text is in italics, the Emphasis Indicator would not be used for this notation.

The Emphasis Indicator carries through spaces, symbols, and other indicators and is in effect until terminated by the End Emphasis Indicator, which must always be used to terminate emphasized notation.

The Emphasis Indicator must be the first symbol when computer notation, displayed or embedded, begins with emphasized letters, numbers, or other computer code symbols.

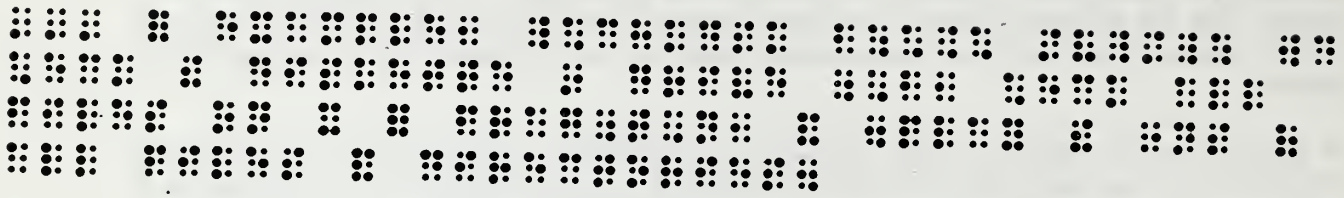
1. [In this example, the underlined terms are the secondary computer notation.]

```
for I := 1 to NoOfLines do
begin
  Readln(Line);
  if Length(Line) < Limit then
    ShortLines := ShortLines + 1
  else
    LongLines := LongLines + 1
end;
```

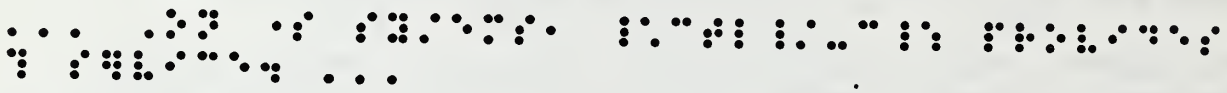


2.

... for example, *chkdsk b:*.* /v* would check the diskette in drive B, check all files on it for fragmentation, and display the names of all files and directories.



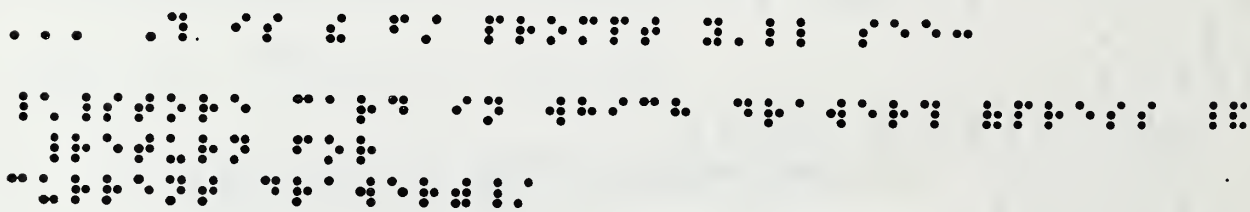
3. for it to finish. [On some systems, *ctl*-c provides this service.] And on some systems, depending on how the terminals are connected, BREAK is a synonym for DELETE or *ctl*-c.



4. SUGGESTION: Screen displays might be able to be done in primary code. Decide when you are setting up the transcription whether the screen displays, prompts, menus, etc., will be more readable in literary rather than computer notation.

When you finish filling out a card, use (ALT) + (F6) to save it on disk. If the cabinet has more than one drawer, you'll need to specify one for storing the card. This is the first prompt you'll see:

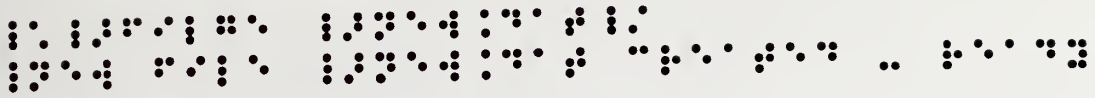
```
Store card in which drawer? (press Return for
current drawer)
```



5. [When you want to create a new file card, choose Make card from the Cardfile Menu.] *Executive Filer* responds with Figure 5.2, the Card Editing screen.

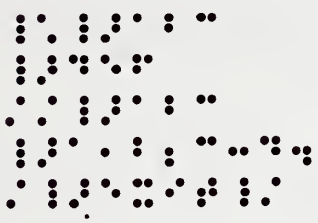
6.

```
FILGE NEW.DAT
New file NEW.DAT created - ready
```



7.

```
ABC
DEF
//ABC
A1B2C3D4
/EXIT
```



The Emphasis Indicator must be used to begin emphasized notation that is a part of computer notation that has begun with any other computer code symbols or indicators.

8.

&H*dddd* where *dddd* is a 1 to 4 digit sequence composed of hexadecimal ...

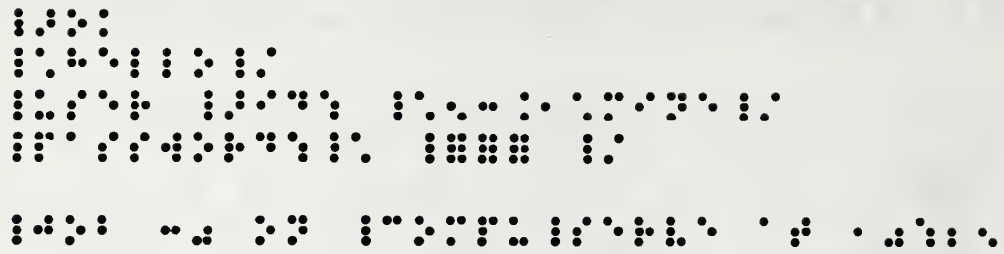
9.

```
$ mail
From doug Sun Sep 25 20:53 EDT 1983
give me a call sometime monday
```

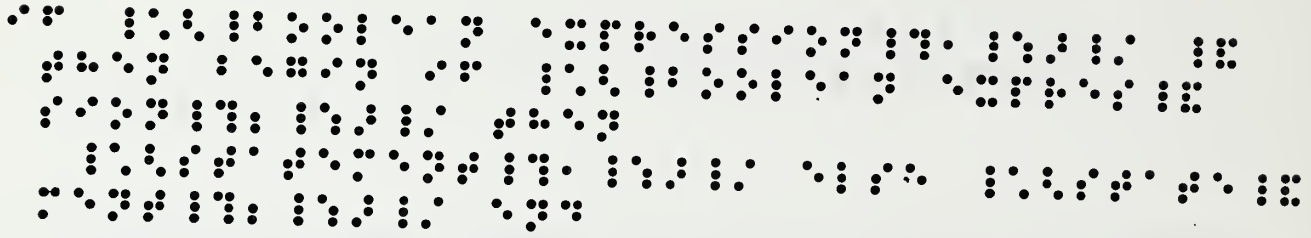
10.

OK
nello
User ID: 53.1*mine
Password:

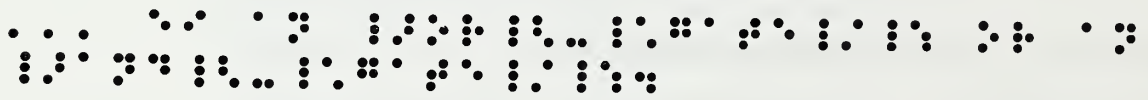
Job 30 on CompuServe at 10:25



11. if ⟨*Boolean expression*$_1$⟩ then begin if ⟨*Boolean expression*$_2$⟩ then ⟨*statement*$_1$⟩ else ⟨*statement*$_2$⟩ end



12. ... either an OR-*gate* or an AND-*gate*.



When the Emphasis Indicator is used in textual passages, the Termination Indicator must be used to indicate a return to the primary code. The Termination Indicator follows the End Emphasis Indicator. (See examples above: 2, 5, 8, 12.)

Be aware of unusual stylistic practices for which the Emphasis Indicator can be used. The problem of transcribing the following examples can be solved by using Emphasis Indicators for the notation that uses key representations.

Suggestion: Use Emphasis Indicators for the single-letter or symbol key notation. Use the Shape Indicator for keys that contain multiple letters or symbols. A transcriber's note should explain this usage.

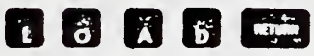
1. 1 Find the SAMPLE PROGRAMS disk and put it into drive 1. Check the catalog by typing

CATALOG,S6,D1

2. 3D0G (that's a zero), or

C600G (these are zeros, too) to retart DOS.

3.



4. If you see the Monitor program's asterisk (*), type

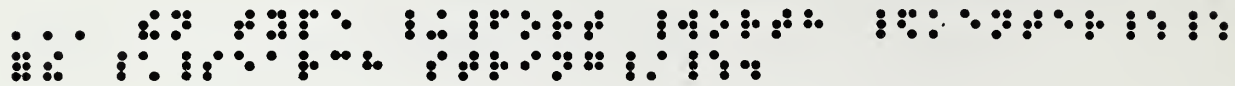
6 CONTROL-P, or

6 CONTROL-K to start DOS.

If more than one different type style is used in print for computer notation emphasis, use one of the Transcriber's Option symbols for the second type style. (See Section XI.)

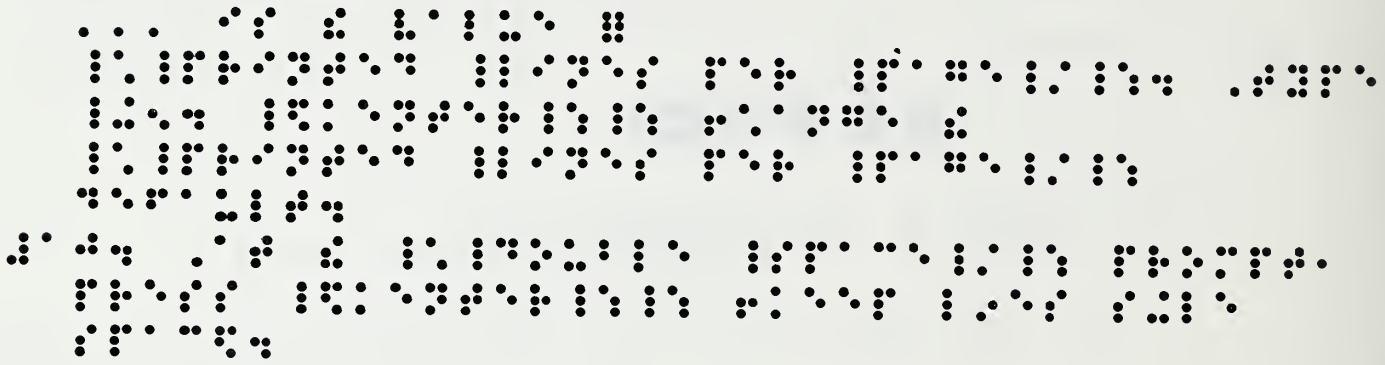
Emphasis Indicator - Additional Examples :

1. Suppose you want to change almost every occurrence of Fort Worth to Dallas. Press F2 for Substitute; then type **Fort Worth** ENTER for the `Search string`.



2. To have the bottom margin also contain 6 blank lines (and the top margin of any subsequent pages), you subtract 12 from 66 and the result, 54, is the value for `Printed Lines per Page`. Type 54 ENTER to change the `Printed Lines per Page` default.

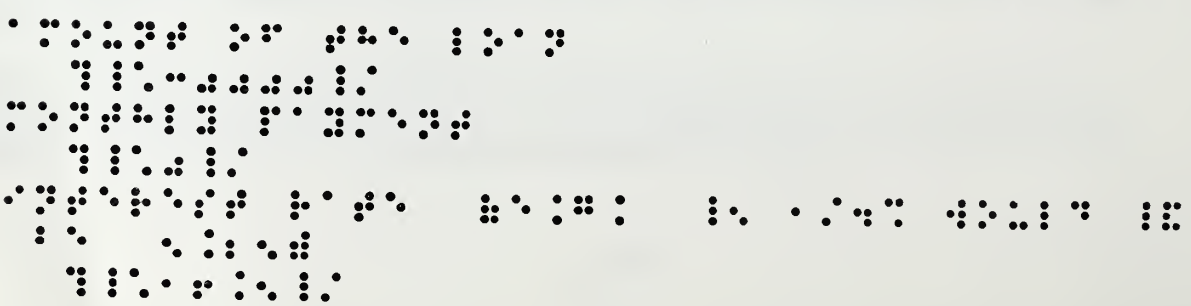
10. At the `Double Space` prompt, press ENTER to keep lines single spaced.



3. [Lighter type is blue in print.]

Blue in examples indicates input from you as opposed to output from the computer.

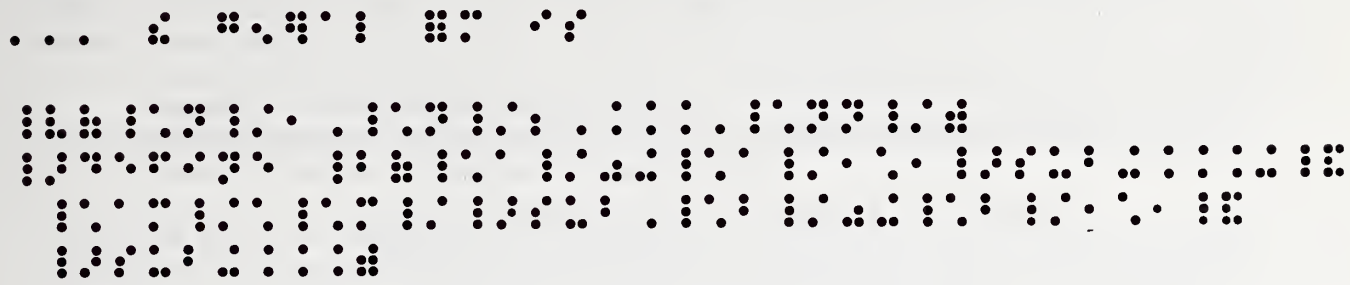
```
ENTER VALUES FOR 3 OF THE 4 FOLLOWING INPUTS
AND A ZERO FOR THE ONE YOU WISH COMPUTED....

AMOUNT OF THE LOAN
 ?30000
MONTHLY PAYMENT
 ?0
INTEREST RATE  (E.G.  5 1/4% WOULD BE  5.25)
 ?16.5
LIFE OF THE LOAN  (E.G. 8 YEARS 2 MONTHS WOULD BE  8.02)
 ?30.0
```



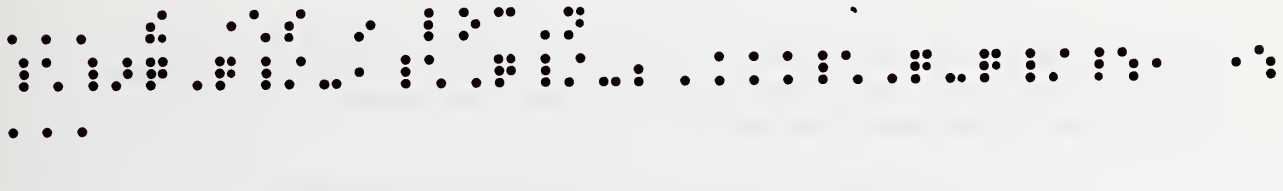
4. In the text containing this example, all lower-case letters in computer notation are italicized. Include a Transcriber's Note at the beginning of the book explaining this and do not use indicators for the lower-case letters.

... the general form is

V(*n*1,*n*2,...,*nn*)
DEFINE B(*a*0+*a*1*1SUB+...+*am**mSUB,*b*0+*b*1*1SUB+...)

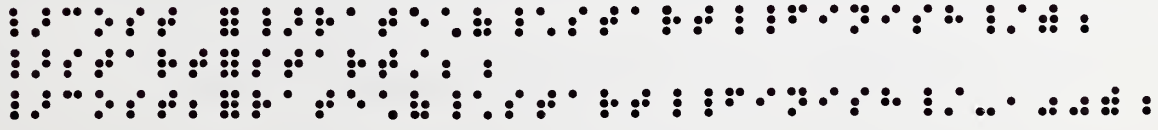


5. ... the characters in location *Q*,*Q*-1,*Q*-2,...,*Q*-*q*, where ...



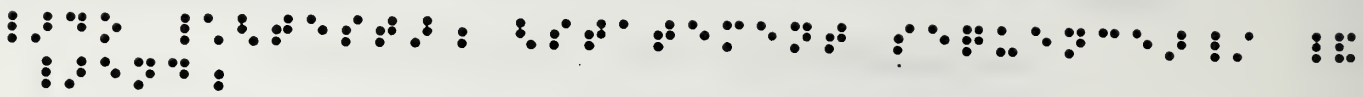
6. Type (P)(R)(SHIFT)-(#)(6) to restart DOS from drive 1, slot 6 on all

7.
```
COST =RATE*(START__FINISH);
START=START/2;
COST2=RATE*(START__FINISH-100);
```



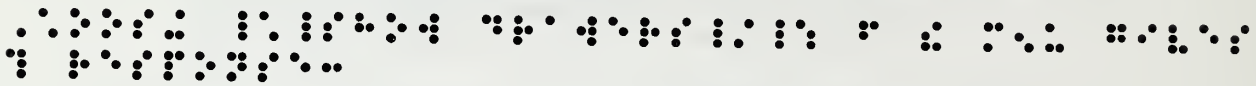
8.

DO ⟨*test*⟩; ⟨*statement sequence*⟩ END;



9. [If all notation from "menus" is literary, the words "Show drawers" in this example could be literary with italics.] (See Section XXIV.)

Choosing Show drawers from the menu gives this response:



10. [Dot matrix is the primary computer notation. Although the type style in embedded text is different, there is no reason to use indicators other than those used for program lines.]

[The next program reads the four fields from FOUR.FRUITS with one input statement. It also displays them to show it's working.

```
10 REM GET.FRUIT WITH ONE INPUT STATEMENT
20 D$ = CHR$ (4)          (D$ is CONTROL-D)
25 PRINT D$;"MON C,I,O"   (Watch the action)
30 PRINT D$;"OPEN FOUR.FRUITS"
                          (Prepare FOUR.FRUITS for use)
```

VII SHAPE INDICATOR
(Section XIII CBC)

Shape Indicators are used to enclose a brief series of symbols describing a shape not represented in the Computer Braille Code. (Also see Section XIII, Paragraph D in CBC.)

The End Shape Indicator must always be used to terminate the shape.

Letters used for the shapes in this section are simply suggestions. You may use any letters you like - be sure to include these on the Special Symbols page. Try to use no more than 4 letters which describe the shape as if they were abbreviations or acronyms. (Eventually we hope a list of specific symbols will be made to be used for most shapes so that transcriptions will be uniform.)

The Shape Indicator must be the first symbol when computer notation begins with a shape, whether displayed or embedded.

1. In this example, spaces have not been left before and after the plus sign because this combination of keys must be pressed at the same time on the computer keyboard. However, if you are unsure of spacing at any time, follow the print spacing. Either way, be consistent throughout the book.

ALT + F6

2. [Open Apple]

or type ☐-CONTROL-RESET

3.

3. Press ALT F3

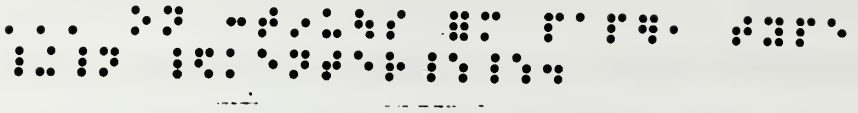
When a shape occurs in textual passages and it is the only or the last term of computer notation, a Termination Indicator must be used after the End Shape Indicator to signal a return to the primary code.

1.

use Block Delete command (CTRL + F8).

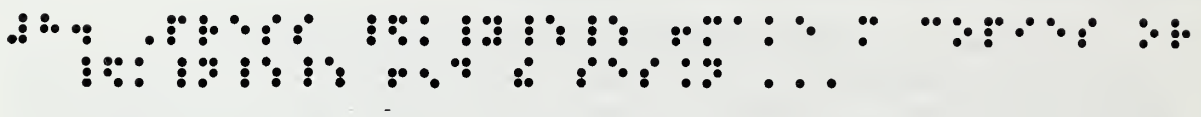
2.

on continuous form paper, type N (ENTER).



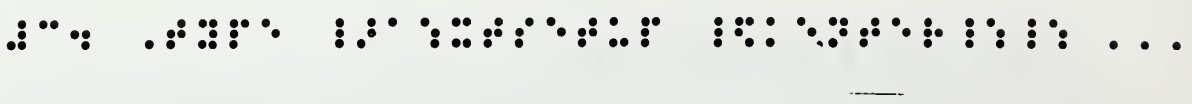
3.

8. Press (Y) to make more copies or (N) to end the session

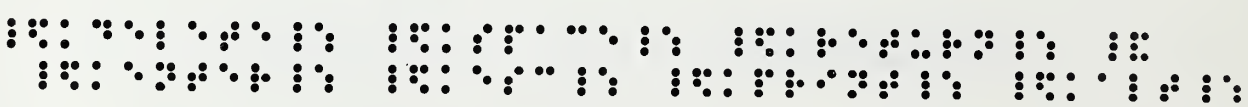


4.

3. Type A:XTSETUP (ENTER)



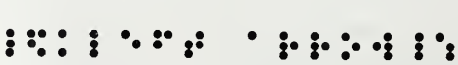
If the shape is a key from the keyboard, follow the Shape Indicator by the letter K and the key legend. Suggestion: When the letters of the keyboard are always depicted in upper-case throughout a text, transcribe without case indicators. A transcriber's note should be included at the beginning of the work explaining this usage.

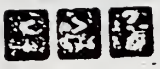


When the key legend contains two symbols, or more, omit any spaces. Transcribe upper symbols before lower ones, left-most before right-most. Use a space if there are two words on a key. (This is not a rule stated in the code, but simply a suggestion. Include a transcriber's note to explain this format.)

1.

(LEFT ARROW)



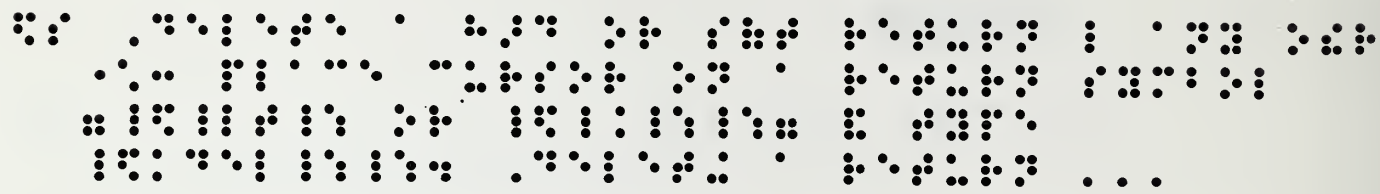
2. 

⠠⠠⠠

SHAPE INDICATOR - ADDITIONAL EXAMPLES

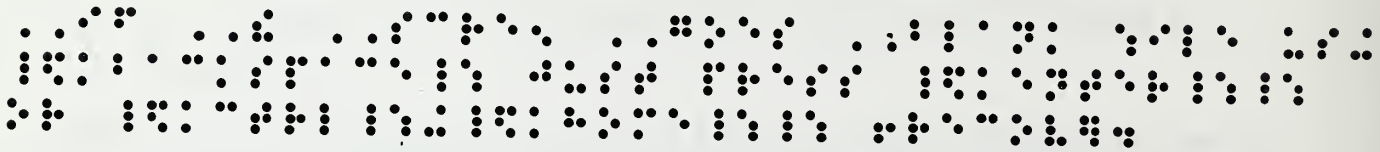
1. [Italics were not necessary here. All such paragraphs are set off by blank lines.]

> [■ *Delete a hard or soft return like any other character: place cursor on a return symbol (◀ or •) and type* DEL. *Deleting a return causes the next line or paragraph to move up.*



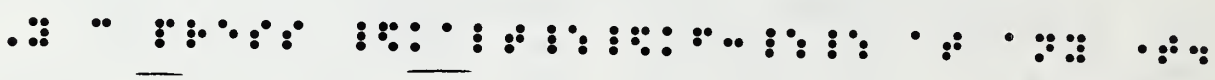
2. [Spaces before and after plus sign, or not. Just be consistent.]

> *IF THE SCREEN GOES BLANK while using* BACKSPACE
> *just press* ENTER *or* CTRL + HOME *to recover.*

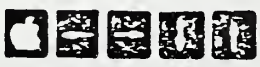


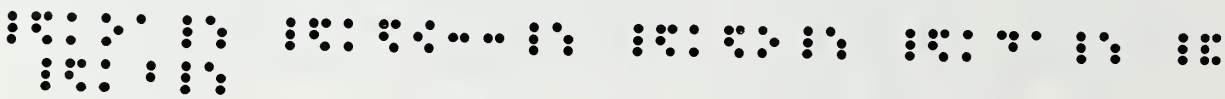
3.

You can press ALT F3 at any time.



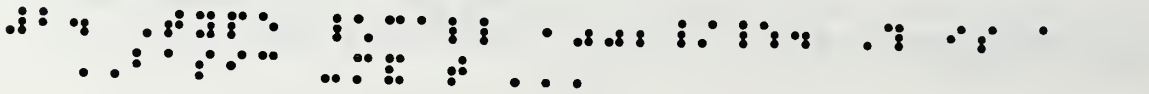
4. [This is a list of keys. Spaces were used to separate them.]



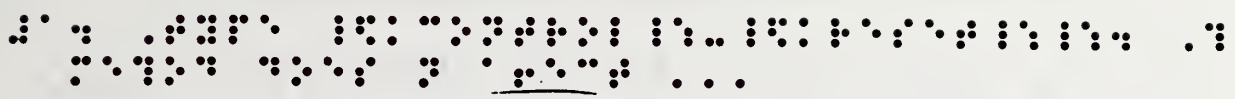


5.

2. Type CALL SPACE 1002. This is a BASIC command that



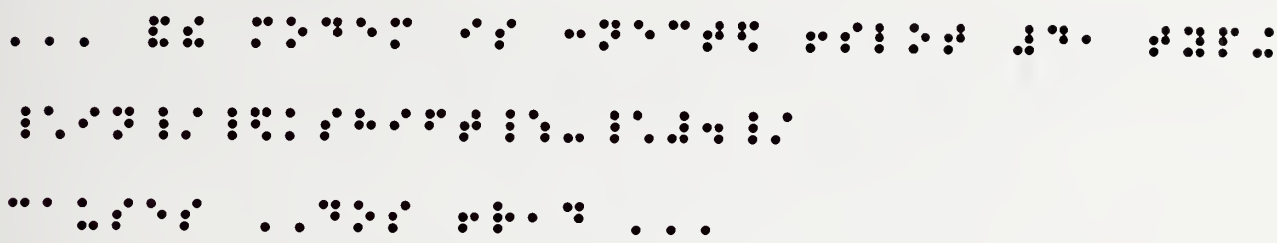
6. 1. Type CONTROL-RESET. This method does not affect



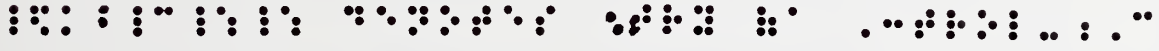
7. 1. If your Apple computer has a telephone modem that connects to a telephone network and the modem is connected to slot 4, typing

I N SHIFT-# 4

causes DOS to read information from the network.



8. CTRL-C denotes entry of a Control-C



9. press CTRL 1 or HOME; then press F2 for Substitute again.

10. line beginning with the symbol ∇. His communication is then with the editor until another ∇ is entered. An example of initial

11. At the Main Menu, press ⊡ to move the cursor over ADDRESS, and press ENTER. The name/address information of Mr. Raymond's customers reappears on the screen.

12. **Tab.** When filling in a menu, Tab moves the cursor forward to the next item; when

13. **Enter.** On any menu, Enter tells Writing Assistant to begin or continue

14. **Backspace.** Moves the cursor to the left one space and erases the character in that location.

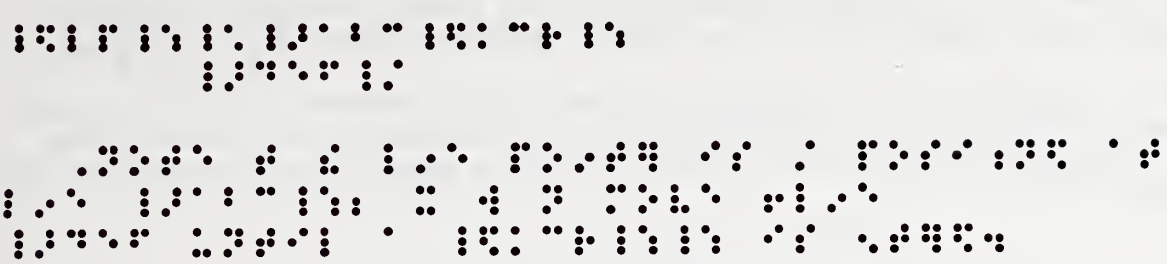
15. [Use any order. Be consistent.]

To clear the screen for action type

16. Or, it may look like this ↵.

17.

Note that the line pointer is still positioned at line ABC; it will not move to line DEF until a CR is entered.

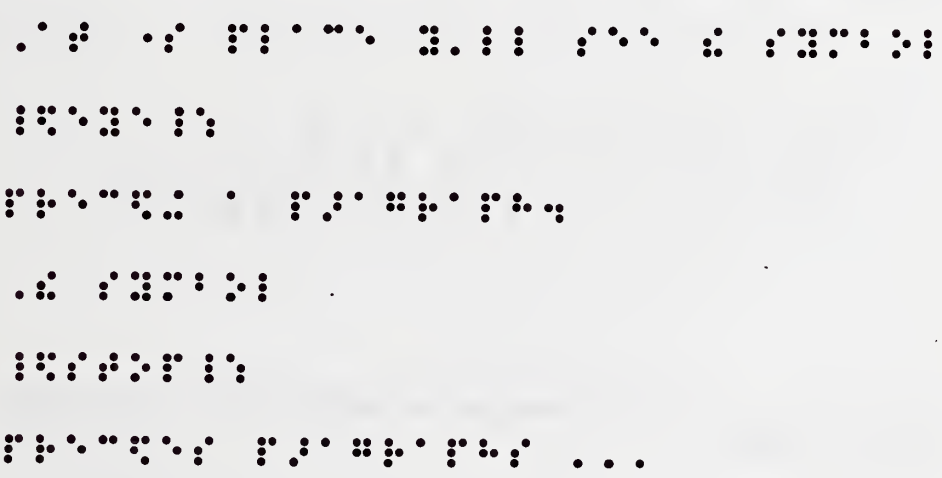


18. At some places you'll see the symbol

preceding a paragraph. This symbol indicates an unusual feature to which you should be alert.

The symbol

precedes paragraphs describing situations from which APPLESOFT may be unable to recover. You will lose your program and will probably have to re-start APPLESOFT.



VIII NEMETH CODE INDICATORS
(Section XII - CBC)

Nemeth Code Indicators are used to enclose purely mathematical expressions, or when a mathematical expression cannot be constructed with computer code symbols. Purely mathematical expressions are often recognizable by a different type style or format from computer notation, or by a knowledge of mathematical notation by the Nemeth transcriber.

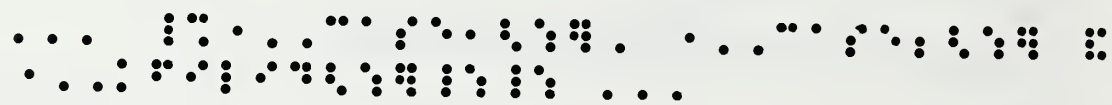
The Nemeth Code Indicator carries through words, spaces, and other mathematical notation.

The Nemeth Code Indicator must be the first symbol used when computer code notation begins with Nemeth Code, whether displayed or embedded.

The End Nemeth Code Indicator must terminate a Nemeth Code term or passage. If this term or passage is followed by notation in the primary code, the Termination Indicator must also be used to signal a return to the primary code.

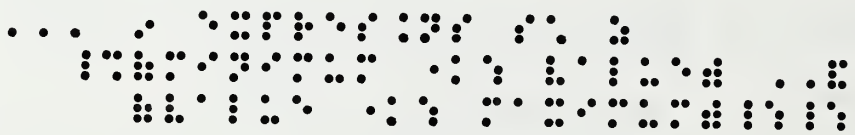
1.

... $\overline{\text{CASE1}}$, $\overline{\text{CASE2}}$ and $\overline{\text{KFILID}}$ ...

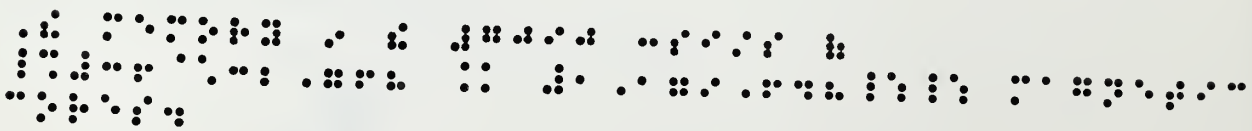


2. From these syntax diagrams you can see why parentheses are necessary in expressions such as

(minimum ≤ value) AND (value ≤ maximum)



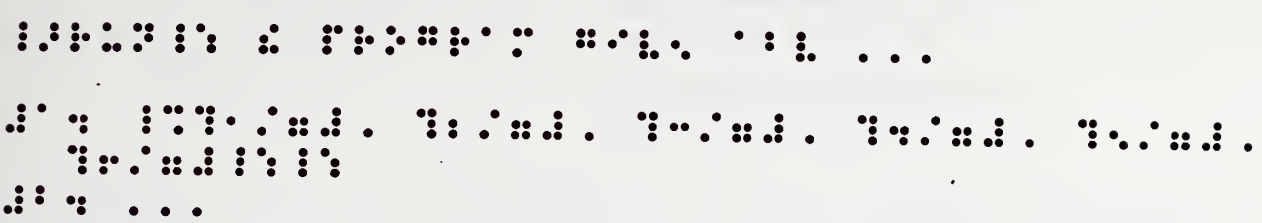
3. **The memory in the 7090 consists of 36 X 32,768 = 1,179,648 magnetic cores.**



4. [Assume #2, which follows #1, is not Nemeth.]

[RUN the program given above] to change each fraction to a decimal. If the decimal repeats, give the block of digits that repeats. Compare the results in each list.

1. $\frac{1}{7}, \frac{2}{7}, \frac{3}{7}, \frac{4}{7}, \frac{5}{7}, \frac{6}{7}$



5.

However, a program can be written that will compute one digit at a time just as you do when you do long division. Then the computer can find as many digits as you wish. [The computation for $\frac{4}{7}$ is given below.]

6. **SETCST LIST ± (RLABEL)**

The End Nemeth Code Indicator returns the transcription to the baseline of writing. Therefore, if a passage that contains Nemeth Code notation has begun with a computer code indicator other than the Nemeth Code indicator, in embedded text, the End Nemeth Code Indicator returns the transcription to computer code and the entire passage must be terminated with the Termination Indicator.

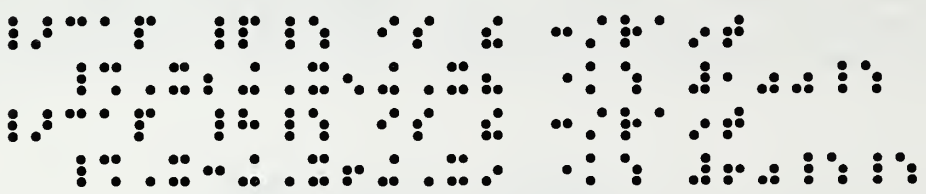
1.

CAP P is the constraint

$$X_2 + X_5 + X_8 \leq 100$$

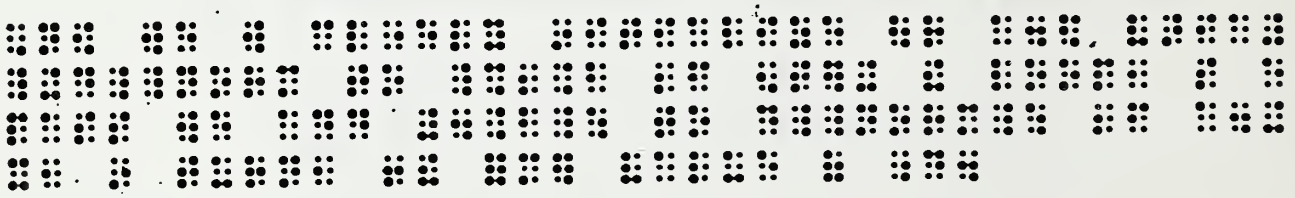
CAP H is the constraint

$$X_3 + X_6 + X_9 \leq 60$$



2.

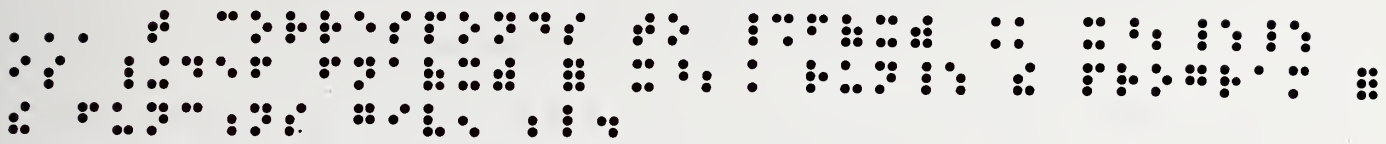
Now we will cleverly initialize our added variable d_norm to 2β before entering the loop, so every pass we add 4β to d_norm before adding it, in turn, to the former value of d.



When a passage that has begun with the Nemeth Code Indicator, in embedded text, is followed by additional computer notation, the End Nemeth Code Indicator must terminate the Nemeth Code and the computer notation which follows must be terminated by the Termination Indicator to signal a return to the primary code.

3. [Italic type has been used for Nemeth throughout the text.]

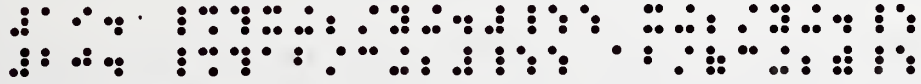
Write a BASIC program to calculate the value of a given function for values of x entered with READ . . . DATA statements. Recall that the BASIC statement that corresponds to $f(x) = x^2$ is DEF FNA(X) = X↑2. RUN the program for the functions given below.



4. Shown below are a number of mathematical expressions and corresponding computer expressions. Each computer expression contains at least one error. Correct each error, rewriting the expression if necessary.

19. $\frac{x+2}{y+4}$; x + 2/y + 4

20. $\frac{ab}{c+2}$; ab/(c +2)



When a reference indicator occurs within displayed Nemeth Code notation, use the Nemeth Code Indicators and Nemeth Code symbols for the reference indicator.

SUGGESTION: Use Nemeth Code Indicators to represent Greek letters in computer notation.

1.

... DOZ is the computation of the pricing vector, Z or π .

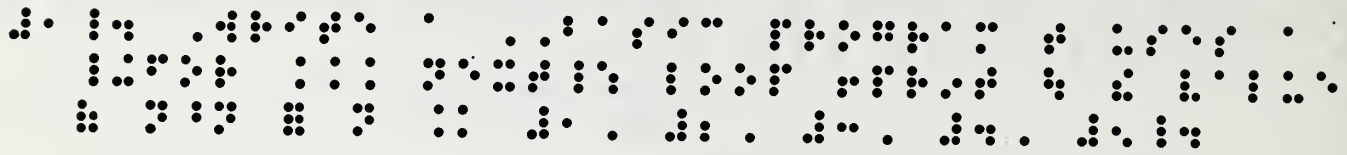
2.

... symbols from the set X including the empty string λ .

When Nemeth Code is the primary code, embedded computer notation should be enclosed in the indicators of the Computer Braille Code.

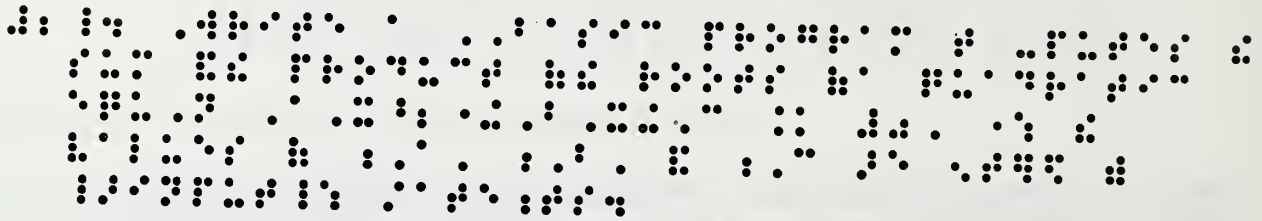
1. [In the BASIC language, letters are exclusively upper case.]

1. Write a BASIC program that uses a FOR . . . NEXT loop to print out the value of n^n for $n = 1, 2, 3, 4, 5$.



2.

2. Write a BASIC program that computes the sum and the product of the roots of a quadratic equation $AX^2 + BX + C = 0$, where the values of A, B, and C are entered with INPUT statements.

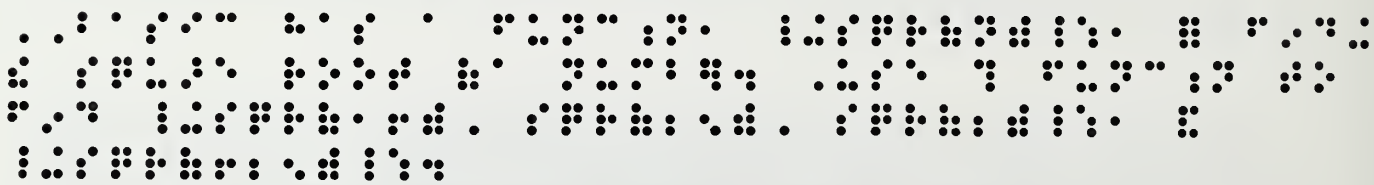


3.

[BASIC has a function, SQR(N), for finding the square root of a number. Use this function to find SQR(16), SQR(25), SQR(2), and SQR(625).

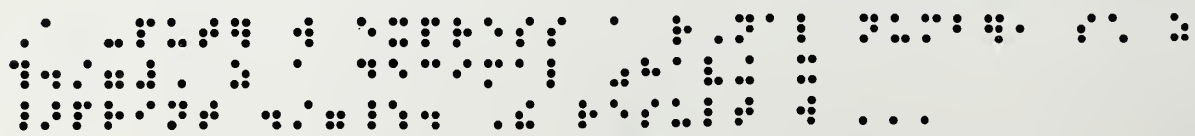
Exercises

1. Write a program to find the distance between (X1, Y1) and (X2, Y2).]
2. Use your program to check your answers to Exercises 1–12 above.



4.

[A computer will express a rational number, such as $\frac{4}{7}$, as a decimal by having it PRINT 4/7. The result will display the fixed number of decimal places for which the computer is programmed.



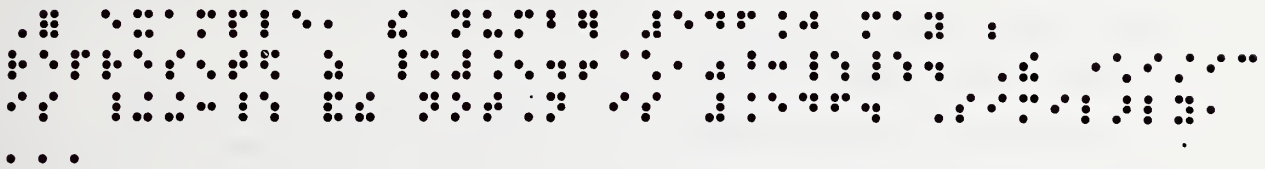
NEMETH CODE INDICATOR - ADDITIONAL EXAMPLES

(Assume Literary/Textbook Format is primary code in these examples.)

1. [This is a displayed expression.]

$$'011010111' \in L,$$

2. For example, the number 546.0 may be represented as $.546 \times 10^3$. The characteristic is +3 and the mantissa is .546. Similarly,



3. [embedded]

$L \subseteq V^*$.

4. [The following has to be a combination of computer notation and Nemeth Code in order to be represented precisely. No spaces before or after equal sign.]

memory word. Each successive term may be given a number, say N, that is one larger than the number designating the previous term. Thus a general way of describing each term would be as follows:

$$\frac{X^N}{N*(N-1)*(N-2)*\cdots*1}$$

[Another way would be the following:

$$t_n = t_{n-1} * \left\lfloor \frac{x}{n} \right\rfloor \quad]$$

IX COMPUTER BRAILLE CODE INDICATORS
(Section XI - CBC)

The Begin Computer Braille Code Indicator is used to begin computer notation in embedded text when that notation cannot be started with any of the four indicators

- Caps Lock
- Shape
- Emphasis
- Nemeth Code

The Begin CBC Indicator carries through spaces, computer code symbols, and other indicators.

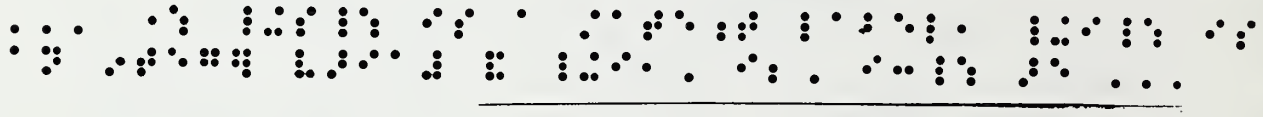
Notation begun with the CBC Indicator must be terminated with the Termination Indicator to signal a return to the primary code. (Also see Section XII - Termination Indicator.)

Use the Begin CBC Indicator when embedded computer notation begins with:

1. Lower-case letters. Do not use the letter sign.

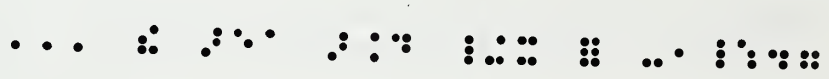
   a. [Italics are computer notation.]

   ... where *s* is a statement label, *i* is an integer variable and *i1*, *i2*, *i3* are ...



   b.

   At times, this function changes very rapidly. Consider how to ensure that the plot is clear everywhere. (Take particular care in [the area around x = -1.)]

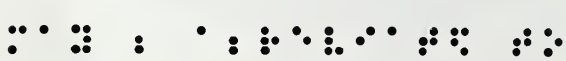


   c. The procedure *writeln* sends a carriage return to the output file, and

```
write(top);
write(bottom);
writeln
```

   may be abbreviated to

```
writeln(top,bottom)
```

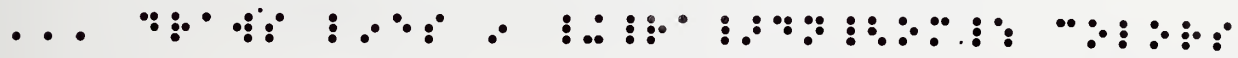


d. ... the three part numbering scheme, aaa.bbb.ccc, to be ...

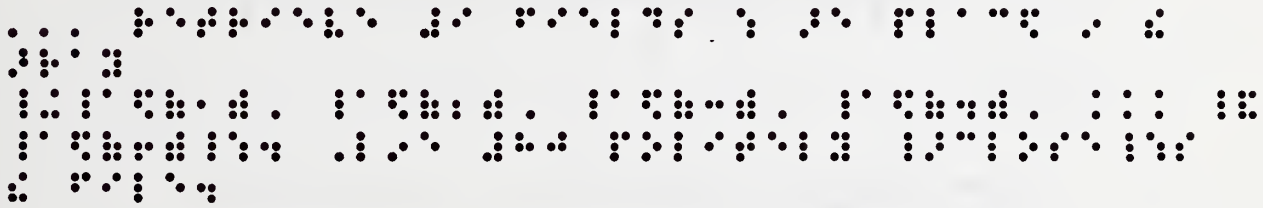
e. [What does ptr_num contain as its value?] If we use the example

2. Single upper-case letters. (See Section IV for a definition of single upper-case letters.) The Shift Indicator must also be used preceding the letter. Do not use the capital sign.

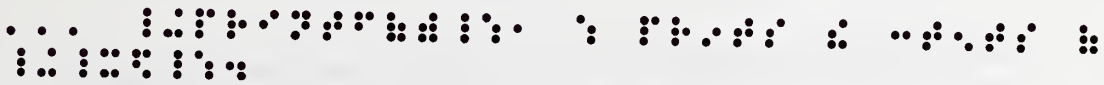
a. [draws lines in RaNDom colors] Can you write a program that across the screen?



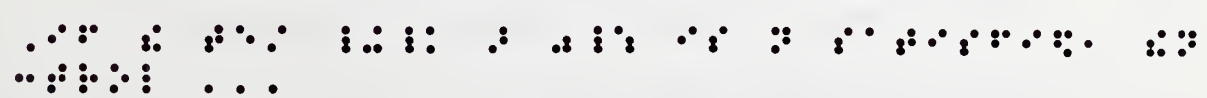
b. retrieve 9 fields which are placed in the array A$(1), A$(2), A$(3), A$(4), ... A$(9). Line 80 politely CLOSEs the file.



c. printf(), which prints the contents of X$.



d. If the test K > 0 is not satisfied, then control

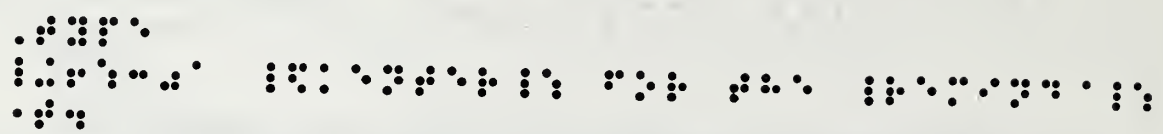


e. of events, the Alarm's beep goes off when an event's Reminder time matches the current time. Also, whenever the date and time

3. Numbers. Do not use the number sign.

a.

Type 6:30a ENTER for the Reminda time.



b.

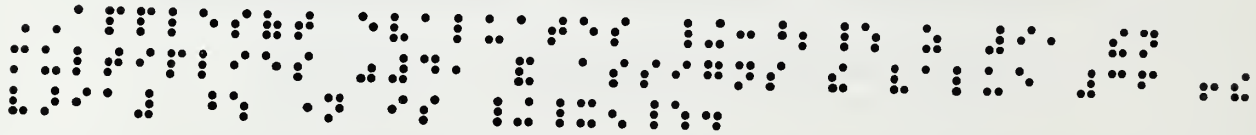
ed begins by reporting the number of characters in junk; the command 1,$p tells it to print all the lines in the file. After you learn how to use the editor,

c.

Change line 30 to 30 LET N = N/10 to see how your computer uses scientific notation for small numbers.

d.

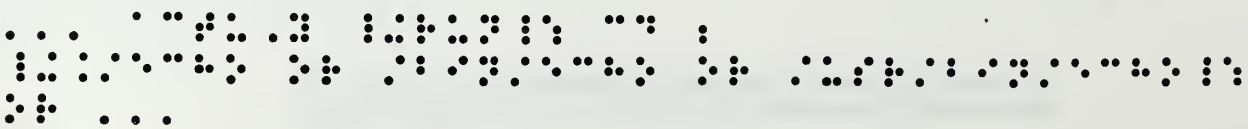
APPLESOFT evaluates 3^2 as 9, then multiplies by 4, and assigns the value 36 to the variable whose name is X5.



4. Other computer symbols

a.

actually run could be ./echo or /bin/echo or /usr/bin/echo or



b.

Suppose we want to initialize a pointer variable to point to **number**. We can do this with the unary **&** operator in C. The syntax is

c. [A subheading - starts cell 5, runovers cell 5]

10. Sample Program Using <string>'s

d.

The message ?SYNTAX ERROR appears.

e.

The shell stores the exit status of the last program in the variable **$?**.

f.

{ } denotes a required entry chosen from among the included items.

g.

in the editor search for a string that can occur anywhere in a line; the special characters ^ and $ are needed to *anchor* the search to the beginning and end

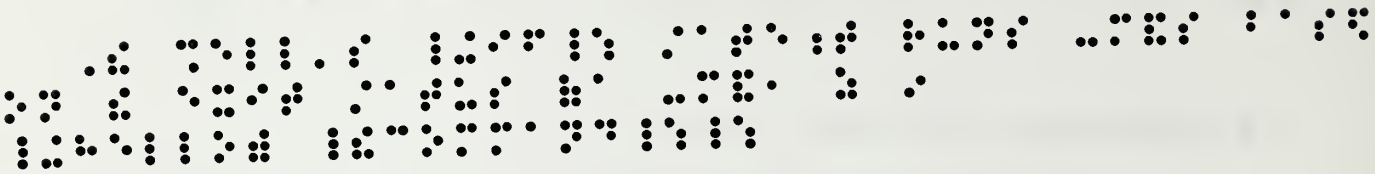
h.

FIRST is made .TRUE. and remains .TRUE. until the (M!)th call.

5. A second emphasized type style that is being represented by the Transcriber's Option Symbol. (The Transcriber's Option Symbols cannot initiate computer notation - See Section XI.)

[In this example, italics are a second emphasized type style.]

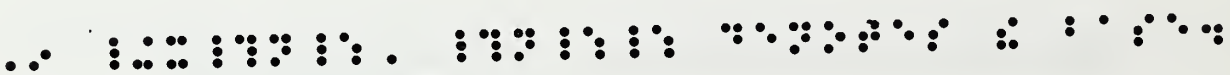
The shell's `if` statement runs commands based on the exit status of a command, as in `hello`) *command*



6. Superscripts or subscripts (This would be a rare occurrence.)

a. In x^2, the ^{2} is the exponent.

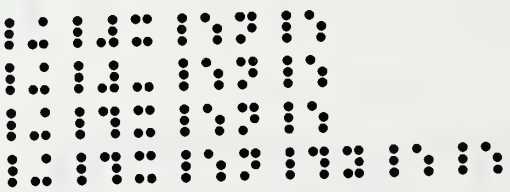
b. In x_n, $_n$ denotes the base.



Assume these examples are embedded:

c. $\underline{x_n}$ d. $\bar{}_n$ e. $_x{}^n$ f. $x^n{}_y$

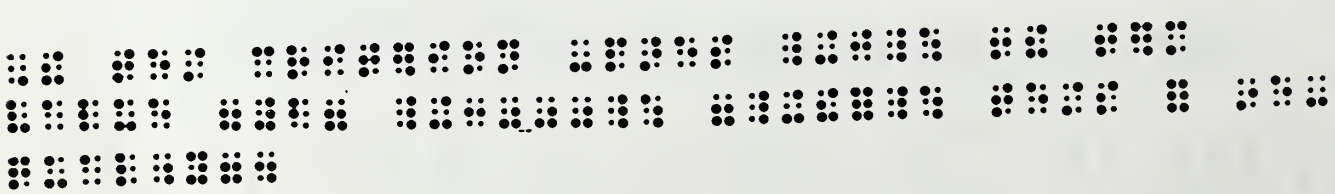
c.
d.
e.
f.



Use the Begin Computer Braille Code Indicator when computer notation in textual passages begins with literary punctuation that is followed by any of the above-noted symbols.

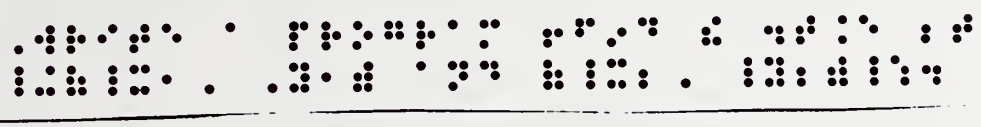
1.

The test criterion compares j to the term value (2) 3000 (!= tests for inequality).



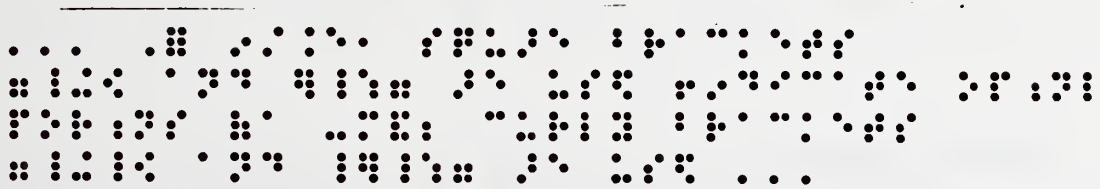
2. these names, and to translate each name when it produces an executable ".CODE" file. These names are called "*identifiers*". In words, the syntax rules state that an identifier may be constructed as follows:

3. Write a program to find the distance between (*X*1, *Y*1) and (*X*2, *Y*2).



4. real root, and two complex roots, are determined once the value of (b^2 - 4*a*c) (on which the appropriate solution depends) has been

5. not specifically discussed in the text. For instance, square brackets ([and]) are used to indicate optional portions of a command; curly brackets ({ and }) are used to indicate those portions that may be repeated. So



NOTE: In many books the author uses enclosing symbols - single quotes, double quotes, parentheses, etc. - to set off computer symbols which are being discussed in the text. For example, you might encounter the following:

several punctuation characters such as '.', ';', ':', '<', and so on,

If this were transcribed according to the rules as stated above for using punctuation in the primary code, the transcription would be:

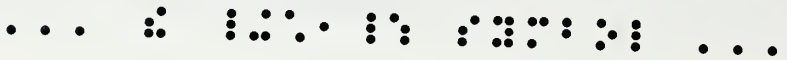
When this device is employed, and it is perfectly clear from the context of the material that these enclosing symbols are not part of any program notation, it is SUGGESTED that the enclosing symbols be omitted in the transcription. A note on the Transcriber's Notes page must explain this procedure.

If enclosing symbols are omitted, the transcription of the above example will be:

ADDITIONAL EXAMPLES

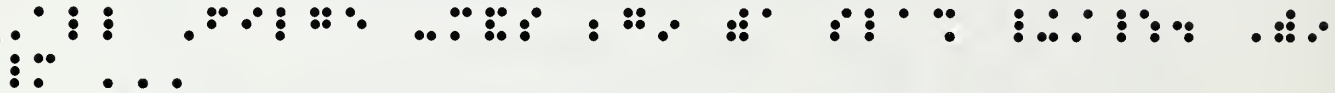
1.

... the "*1" symbol ...



2.

All Filge commands begin with a slash (/). Within many



3.

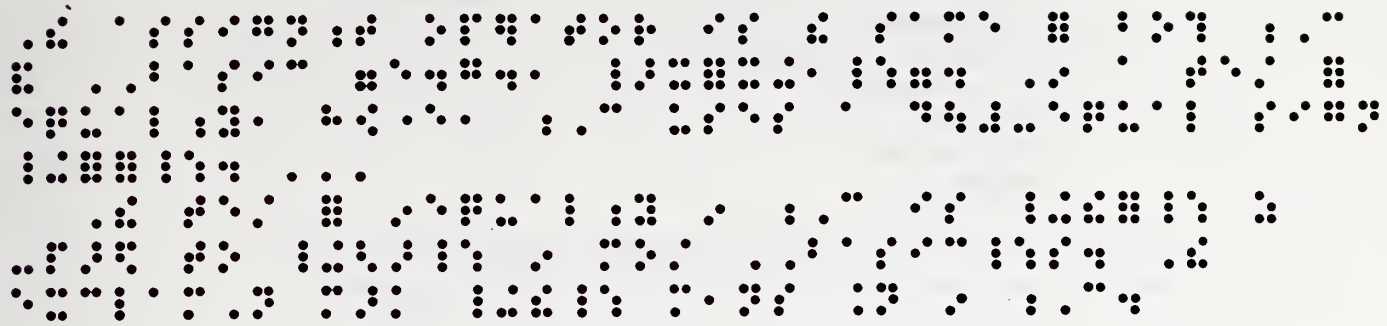
The syntax for <Boolean expression>, Figure 3-5, shows that there is an order of precedence similar to that which applies to the precedence of the multiply operator ("*") over the addition ("+") and subtraction ("-") operators. Thus NOT is executed before AND, while AND is executed

[Assume this line is embedded.]

```
140 LET P=P+A(J)*X↑J
```

4. [The assignment operator is the same for both C and BASIC (e.g., X=X+1). In a test for equality, however, C uses a double-equal sign (==).] With different operators, you can tell at a glance which operation is being performed.

The test for inequality in C is != as compared to <> in most BASICs. The exclamation mark (!) means *not* in C.



Use the Begin CBC Indicator to clarify a potentially ambiguous notation.

```
item.s
```

Although CBC indicators are never used in displayed computer notation that is set off from the body of the text by blank lines, it may be necessary to use them in notation that is displayed according to the rules of the primary code.

1. When the software system starts running, several lines will appear on the computer's display screen [including the following

```
Command: E(dit, R(un, F(ile, C(ompile, X(ecute
U.C.S.D. Pascal SYSTEM II.3B
```

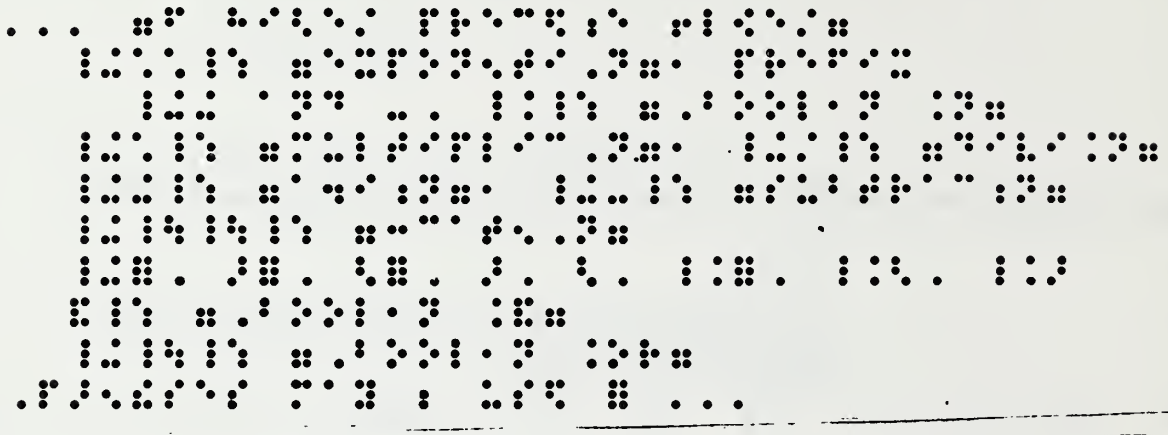
]

2. [Transcriber's Option symbol assigned to Boolean not. (The first Transcriber's Option symbol has been used elsewhere.)]

operators. Other primitives are invoked using the syntax for a programmer-defined function call. The prefix and infix operators have the hierarchy [(from highest precedence to lowest)

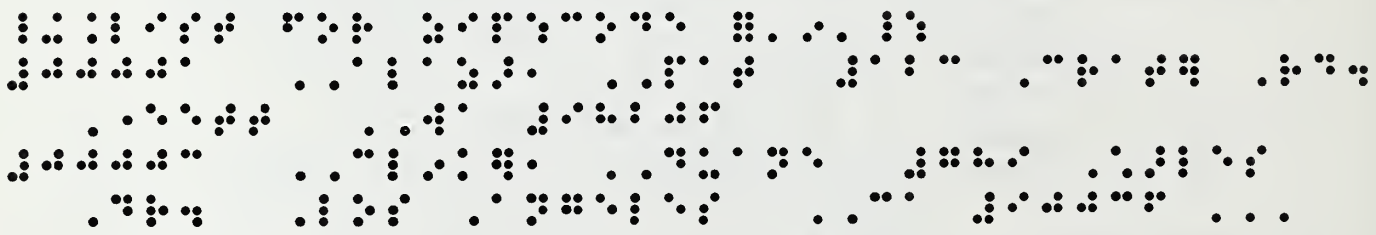
** (exponentiation), prefix + and - , ¬ (Boolean *not*)
* (multiplication), / (division)
+ (addition), - (subtraction)
|| (concatenation)
=, >=, <=, >, <, ¬=, ¬<, ¬>
& (Boolean *and*)
| (Boolean *or*)

Parentheses may be used for explicit control within expressions in the usual way. Because optimization of the order of expression



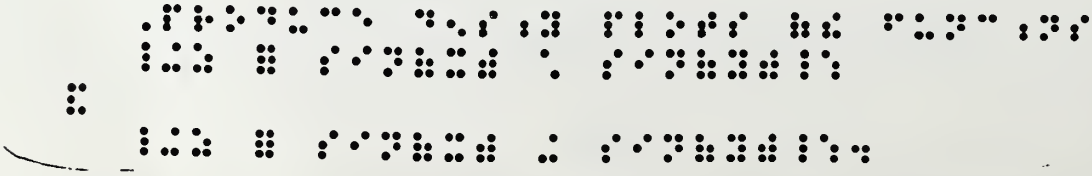
3.

```
. list for zip:code ='9'
00001   ALAZAR, PAT          123 Crater Rd.      Everett        WA 98206
00003   CLINKER, DUANE       789 Charles Dr.     Los Angeles    CA 90036
00005   EMBRY, ALBERT        345 Sage Avenue     Palo Alto      CA 94303
00007   GREEN, TERRY         567 Doheny Dr.      Hollywood      CA 90046
```



4. Produce density plots of the functions

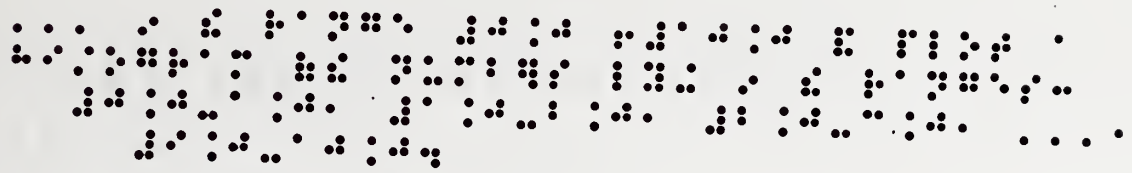
```
    z = sin(x) * sin(y)
and
    z = sin(x) + sin(y).
```



5. [This can be literary!]

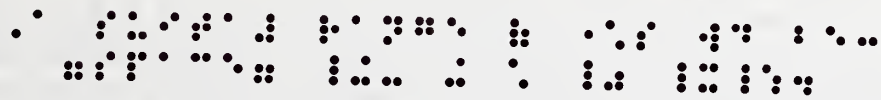
11.5 Write a program to read in 100 floating point numbers [in the range 0.0 to 10.0 and plot a histogram of the numbers lying in the ranges:

0.0-1.0, 1.0-2.0, 2.0-3.0, ..., 9.0-10.0.]



6. A suitable range of characters would be:

(space) - + * 0 X.



When it has been determined - and so stated in a Transcriber's Note at the beginning of the work - that computer notation throughout is upper-case, the Computer Braille Code Indicators will be used for computer notation in textual passages that begin with upper-case letters. (For more information on exclusively or predominantly upper-case transcriptions, see Section XVIII.)

1. Compare the example program GRAPH1 with this diagram and satisfy

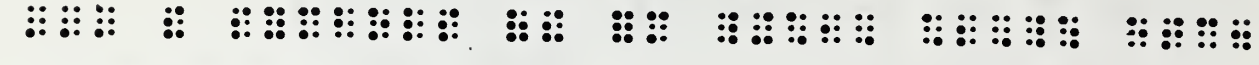
2. likely, the first two items would be read from files. This situation [can be represented by using READ and DATA statements] in the program as explained

3. in programming languages. Commonly statements begin with a key word [designating the statement type: READ, IF, GOTO, etc. Others]

The Computer Braille Code cautions against frequent switches between computer notation and primary code notation, particularly within the same line.

1.

... the symbols of the form *1, *2, etc.

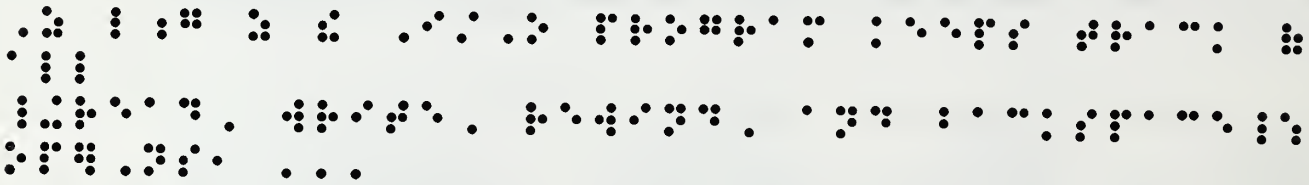


2.

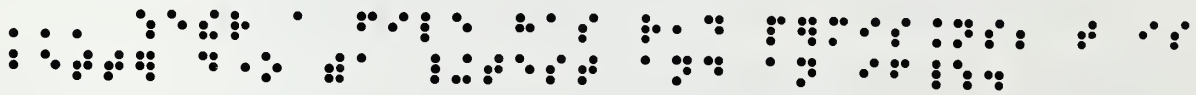
... the symbols of the form *1 and *2, etc.

NOTE: In the examples above, an asterisk is shown. Be aware that the asterisk is not in superscript position. The asterisk prints in this position when the type is done on a typewriter and on some printers.

3. **As long as the I/O program keeps track of all READ, WRITE, REWIND, and BACKSPACE operations, ...**



4. whether a file has read permissions; that is better done with a test and an if.

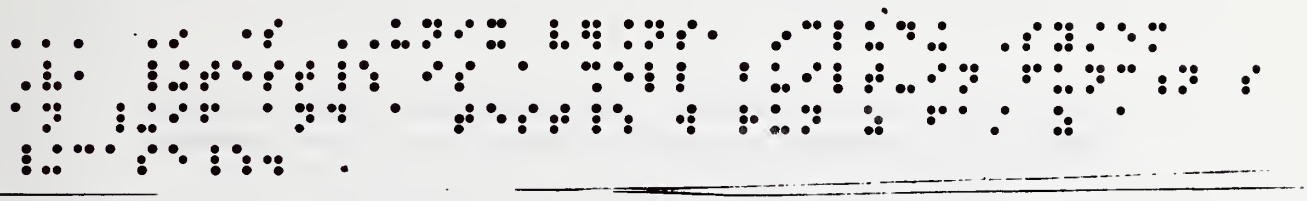


5. In Figure 1-1, the word COMMAND represents any DOS command (for example, SAVE). The fn, [.Sn], [.Dn], and [.Vn] are the command's arguments. An argument in square brackets is optional.

1. sample program described in Section 10 of this chapter.) (To start the TURTLE program from the "Command:" level, press the "X" key to eX(ecute, (the "E" is reserved for E(dit), then type:

TURTLE)

2. **case** statement does pattern matching directly in the shell. (In some UNIX versions, including System V, **test** is a shell built-in function so an **if** and a **test** will run as fast as a **case**.) If **test** isn't built in, **case** statements are more efficient than **if** statements, and should be used for any pattern matching:

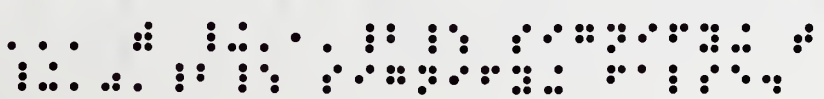


3. **The macro NEXTM used in the code pattern for + can be defined as follows:**

4. **Contents listings are 1 (ON), 0 (OFF), and -1 to -15.**

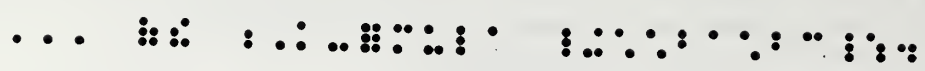
5. [Quotes are part of computer notation.]

... with '1'B signifying true and '0'B signifying false.



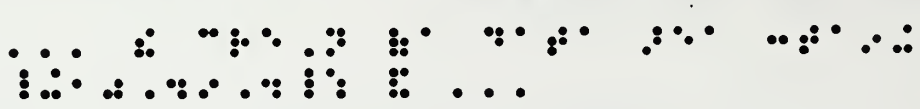
6.

of the K-formula **ba*bc.



7. [It is not certain whether spaces should be inserted after commas - follow print.]

... the creation of a data area containing 10,49,4 and ...

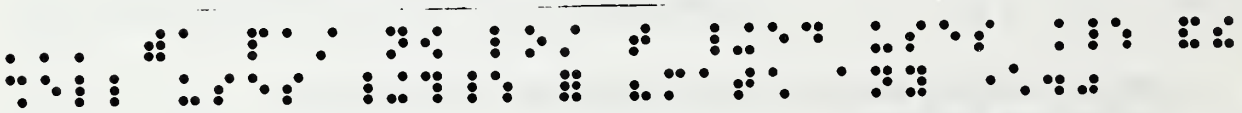


8.

... if x = y, then ...

9.

with a past now lost that ed uses '.' and the shell uses '?' for "match any character." But sometimes the patterns do different jobs. Regular expressions

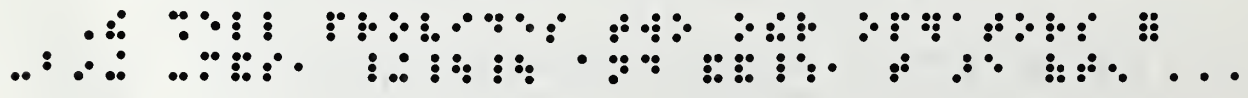


10.

[ ] denotes an optional entry. If multiple items are stacked within brackets, you may enter one of them or none. For example, **TYP[E]** means that you must enter **TYP** but the E is not required.

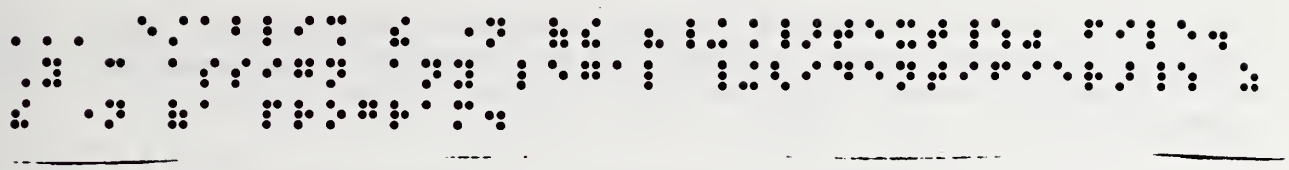
11. | is called a vertical bar in print.

The shell provides two other operators for combining commands, || and &&, that are often more compact and convenient than the `if` statement. For example, || can replace some `if` statements:



12.

establish the name of the ".TEXT" file. You can assign any legal <identifier> as the name of a program.



13.

files (The `stdio.h` file is covered in Chapters 5 and 8)

14.

.The RETURN statement in `printf()` sends control back to `main()` for further processing. Because there are no further statements in `main()`, the program

15.

obvious where one statement ends and another begins. A semi-colon (;) is used for this purpose. This therefore is the heading for our first program, which shall be called 'first':

16.

strings which fit in neither of these categories (for example, the FORTRAN .EQ. and **).

17.

The * is not limited to the last position in a filename — *'s can be anywhere and can occur several times. Thus

X HALF-LINE SHIFT INDICATORS
(Section XV - CBC)

The Half-Line Shift Indicators are used to indicate superscripts and subscripts in expressions which can otherwise be transcribed in computer notation. (See also Nemeth Code Indicators, Section VIII for mathematical expressions.)

The Half-Line Shift Indicator is terminated by a space if the space is followed by computer notation.

1.

... A listener may set NDAC low (shown between t_5 and t_6) but,

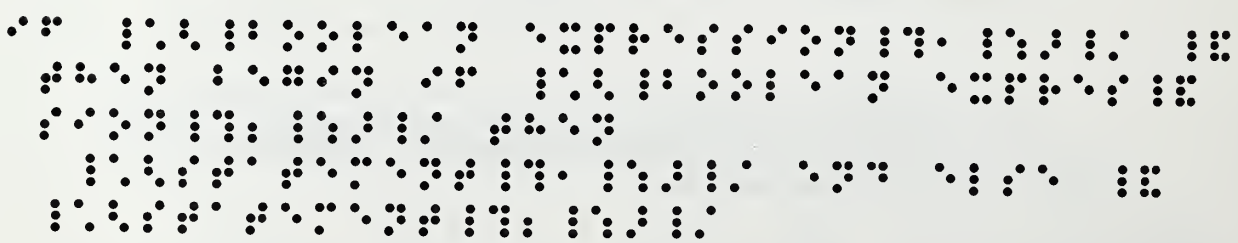
it must do so before NRFD is set high at t_6.

2. In BASIC

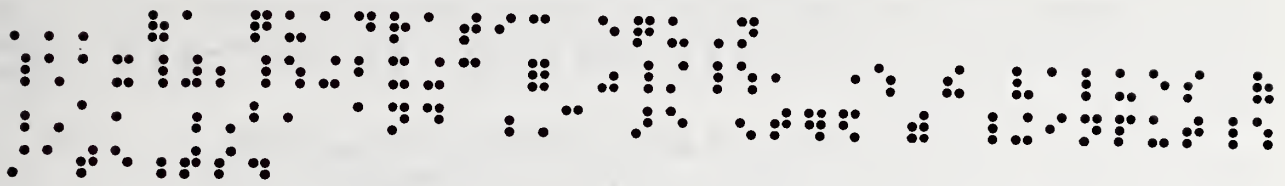
$\times 10^3$ is written E+03 and $\times 10^{-3}$ is written E−03.

The Half-Line Shift Indicator is terminated by the End Half-Line Shift Indicator if it is followed by computer notation with no intervening space.

1. if ⟨*Boolean expression*$_1$⟩ then begin if ⟨*Boolean expression*$_2$⟩ then ⟨*statement*$_1$⟩ end else ⟨*statement*$_2$⟩



2. Write a BASIC program that computes the value of the discriminant of a quadratic equation $AX^2 + BX + C = 0$, where the values of A, B, and C are entered with INPUT statements. The program should then report the number of real roots of the equation.



3. '$01^+01^+...01^+$'

4. /KEYWORD/string$_1$/string$_2$

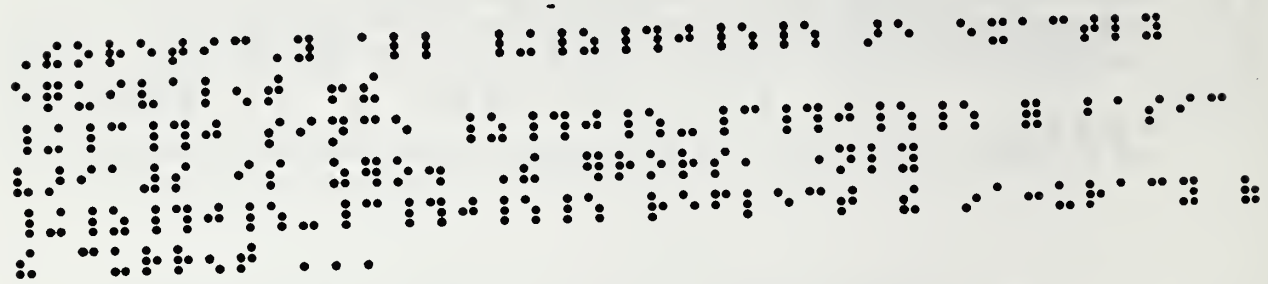
The Half-Line Shift Indicator is terminated by the End Half-Line Shift Indicator if it is followed by notation in the primary code. (In this case at least one Termination Indicator will follow the End Half-Line Shift Indicator, since the Half-Line Shift Indicator will always be preceded by either computer notation or another indicator.)

1.

This number divided by $\emptyset FF_{16}$ is the fraction of the symbol table used.

2.

Theoretically all Z_j are exactly equivalent to the C_j since $Z_j - C_j$ for basic variables is zero. The errors, namely $Z_j - C_j$ reflect the inaccuracy of the current



3.

Study the data in Exercises 1 and 2. For a given value of n, which appears to be larger, $n!$ or n^n? Can you explain why?

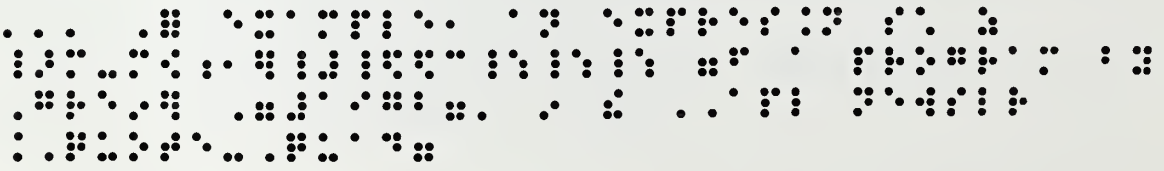
In the rare event of a superscript or subscript occuring as the first symbol of computer notation in embedded text, the Computer Braille Code Indicator must begin the notation, followed by the Half-Line Shift Indicator. (The Half-Line Shift Indicator cannot initiate embedded computer notation.) (See Section IX - #6 in this presentation for examples.)

For reference indicators designating footnotes, see Section XV - c. in the Computer Braille Code.

In another possible occurrence, if a superscript or subscript is a shape, and this is the last term of embedded notation, the notation would need an End Shape Indicator, and a Termination Indicator to signal the return to the primary code.

Devised example --

> be compressed into a single expression. For example, an expression such as M-M[;1]° (from a program by Greiner [1972] in the APL newsletter *Quote-Quad*)



In the following example, lower-case letters have been italicized throughout the book and are therefore considered regular computer notation. The use of literary single letters in embedded text is considered to be less difficult to read because it cuts down on the large number of symbols that would be needed for computer notation and because it would cause no problem in understanding by the reader.

7.2.2 Data Initialization Statement. A data initialization statement is of the form:

DATA $k_1 / d_1 /, k_2 / d_2 /, \ldots, k_n / d_n /$

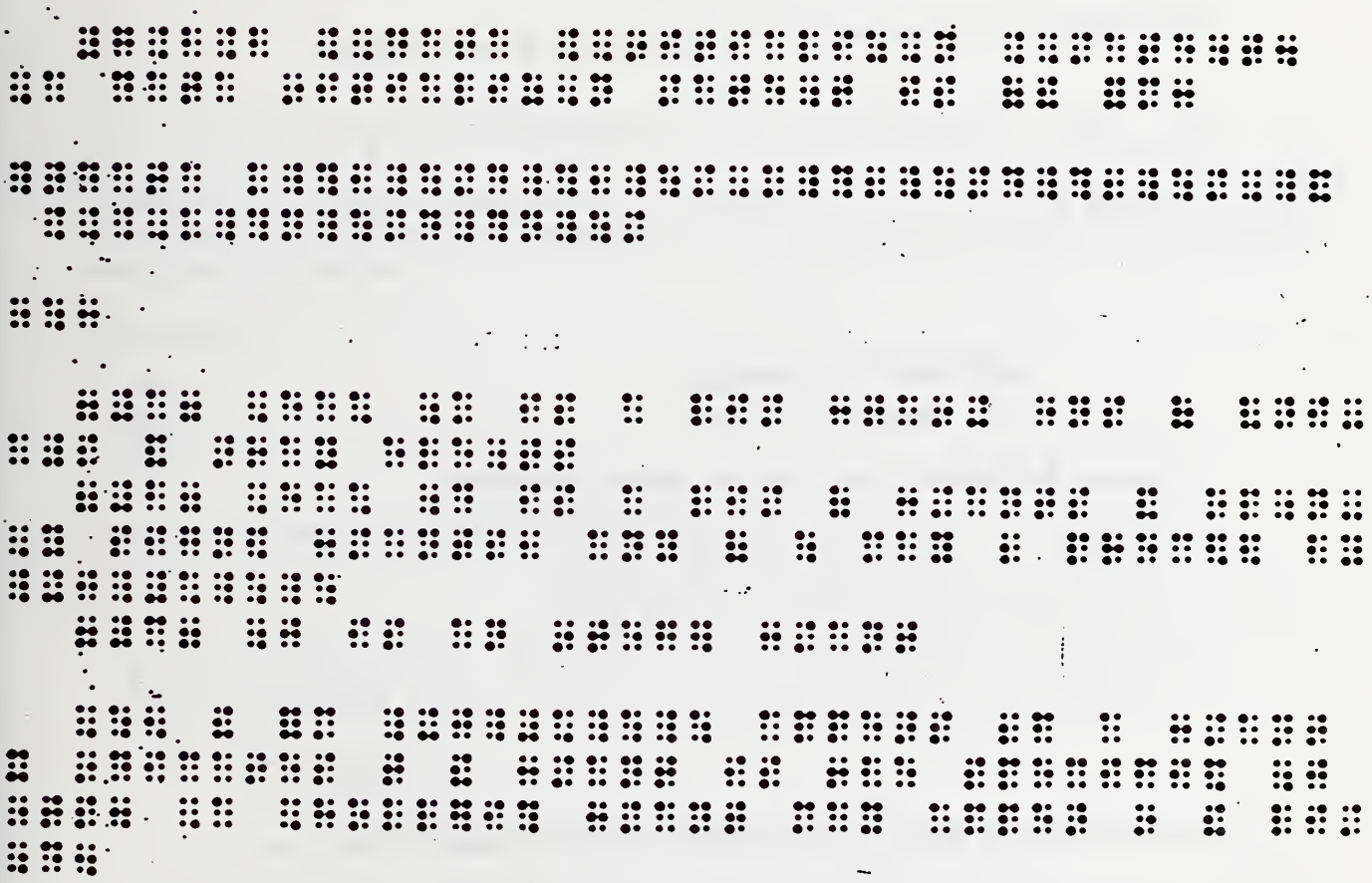
where:

(1) Each k is a list containing names of variables and array elements

(2) Each d is a list of constants and optionally signed constants, any of which may be preceded by j^*

(3) j is an integer constant

When the form j^* appears before a constant it indicates that the constant is to be specified j times. A Hollerith constant may appear in the list d.



XI TRANSCRIBER'S OPTION SYMBOLS
(Section XIV - CBC)

Read Section XIV in the Computer Braille Code.

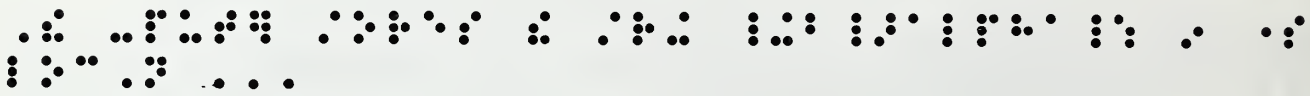
Make a list of computer symbols used in the work you will be transcribing. Such a list is often included in the book somewhere. Look through the pages of the book for unusual symbols and determine if they are used often. If a symbol is used extensively, assign one of the Transcriber's Option Symbols to it. If a symbol occurs in the text that is not included in the CBC and a symbol in CBC does not occur in the text, substitute the two symbols for one another. A note on the Transcriber's Notes page should explain the use of Transcriber's Option Symbols.

Some examples are given for this use of the Transcriber's Option Symbols in Section VII. Some additional examples follow here.

1. [⠘ represents a space.]

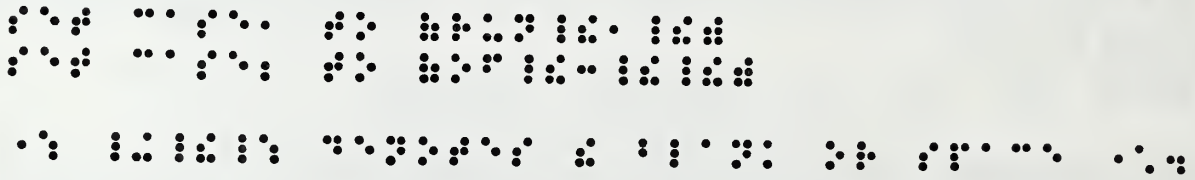
Let the first two cards contain
⠘126⠘327.02⠘-9217⠘⠘3091⠘
⠘ALPHA⠘68.27.6*X-Y

2. **The computer stores the string bALPHA in some location ...**

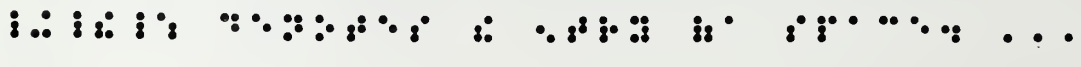


3.
SET CASE1 TO (RUN△1△)
SET CASE2 TO (OF△3△△)

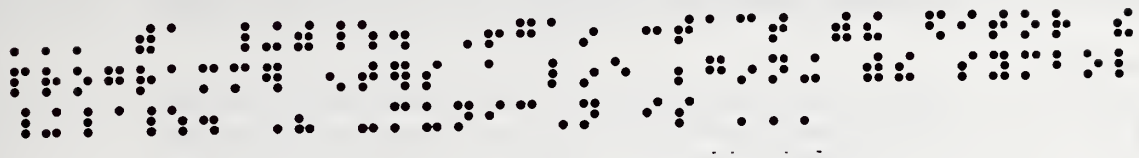
where △ denotes the blank or space character.



4. [▢ denotes the entry of a space.] This symbol is used where needed for clarity only.



5. [with a). To gain contact with the editor the programmer enters a line beginning with the symbol ∇. His communication is then with the editor until another ∇ is entered. An example of initial subprogram entry and editing is given in Section 16-3.



6.

In addition to precisely selecting data from your database, the LIST command can be used to provide you with system information.

[♦LIST STRUCTURE♦ shows you the structure of the database in USE.]

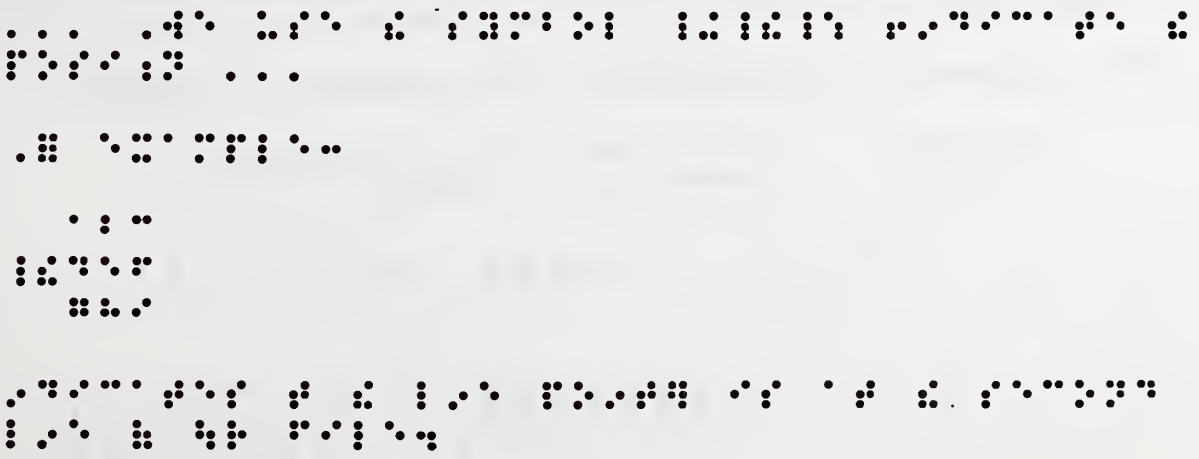
6. [First Transcriber's Option used for this symbol.]

quickly learn to follow its position and to manipulate it as you work with your file. We use the symbol ► to indicate the position of our line pointer as we work with the examples throughout this manual.

For example:

```
  ABC
►DEF
  GHI
```

indicates that the line pointer is at the second line of our file.



7. Second Transcriber's Option used for this symbol, which is in the same book as the symbol above.

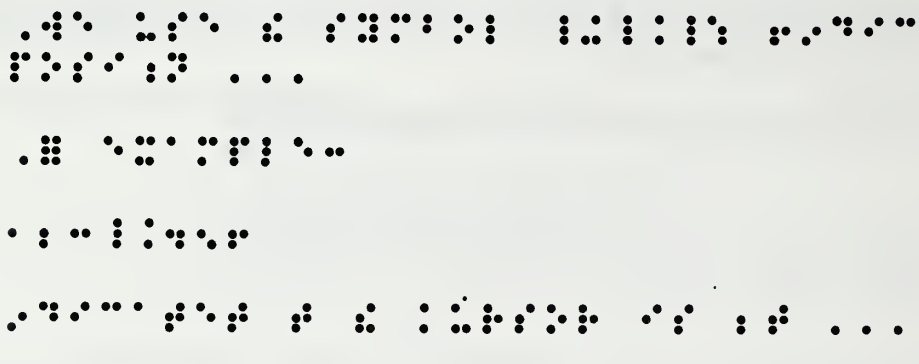
kursor

similar to the line pointer, except the kursor is a pointer between 2 text characters on the current line. We use the symbol ▲ to indicate the position of our kursor in examples contained in the advanced section of this manual.

For example:

123▲456

indicates that the kursor is between the third and fourth characters on the current line.



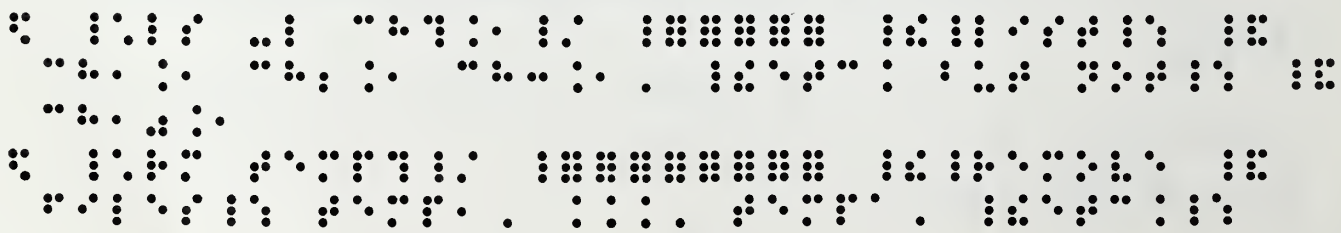
When more than one type style is used in computer notation for emphasis, use one of the Transcriber's Option Symbols to indicate the second type style. When used for this purpose, the rules follow those for the Emphasis Indicator, with the exception that Transcriber's Option Symbols cannot begin computer notation in embedded text. If the second emphasized type begins computer notation in embedded text, use the Computer Braille Code Indicator to begin computer notation followed by the Transcriber's Option Symbol, then the emphasized text.

CAUTION: Do not assign the same Transcriber's Option Symbol to both a symbol and an indicator.

The termination indicator should be used to end this emphasized type, just as the End Emphasis Indicator is used. Consequently, when the material affected by the Transcriber's Option Symbol ends notation in embedded text, the Termination Indicator is used to end the emphasized material and another Termination Indicator is used to signal a return to the primary code.

1. [First emphasis is for slanted letters; second emphasis is for italics.]

`$ ls -l ch?.1`	*List* `ch1.1 ch2.1 ch3.1`, *etc. but not* `ch10.1`
`$ rm temp?`	*Remove files* `temp1, ..., tempa`, *etc.*



2. [First emphasis is for notation represented by keys; second emphasis is for dot matrix type.] Note that if all dot matrix type is used for screen displays, and all screen displays are literary words or sentences, literary notation may be used for these displays.

Any type of file may be verified. If the file does not exist, you'll get a FILE NOT FOUND message.

The slot number, drive number, and volume number arguments are optional. You only need to specify a number when you don't want to use the default. These arguments are described in Chapter 2.

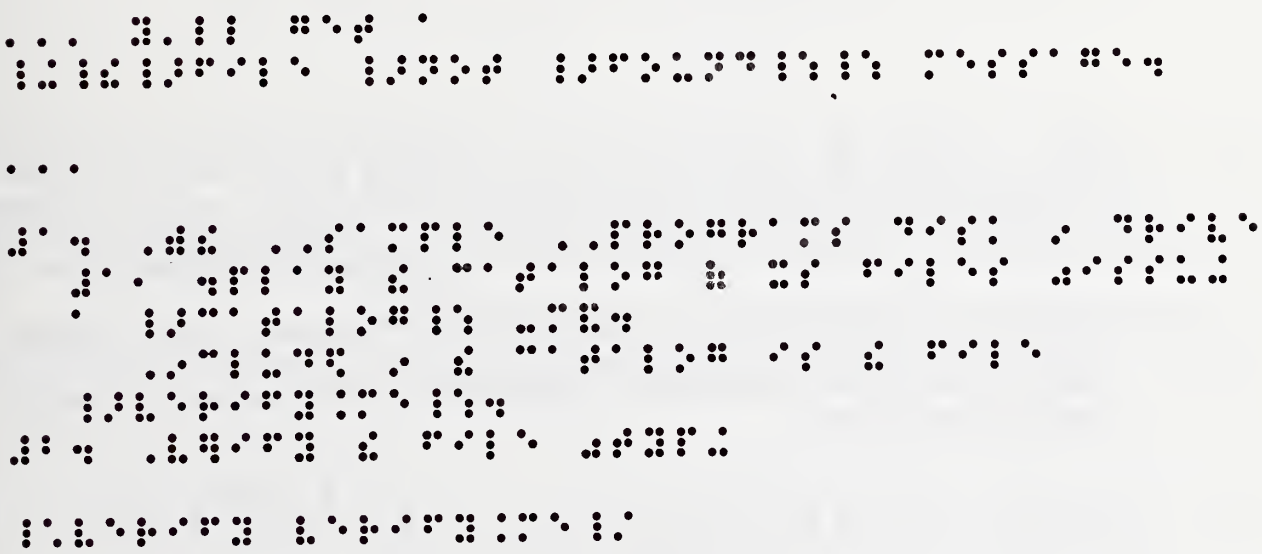
Example

1. With the SAMPLE PROGRAMS disk in drive 1, display the catalog of its files by issuing a CATALOG command.

   Included in the catalog is the file VERIFY.ME.

2. Verify the file by typing

   VERIFY SPACE VERIFY.ME



3. [Primary emphasis is typewritten style type (not shown here). Second emphasis is regular print upper-case letters.]

> 3. With the A> prompt on screen, place the *Executive Filer* disk in drive A.
>
> 4. Type COMBINE ENTER
>
> Follow the instructions on screen. When the A> prompt reappears, the *Writer/Filer* disk is ready to use.
>
> Store the *Executive Writer* disk safely. If you use a different printer in the future, repeat step 1, then copy the files to IDP.$CF and PRTDRIVE.COM from the *Executive Writer* disk to the *Executive Filer* disk.

[illegible]

4. [If slanted letters were used for a second emphasis]

> $ rm * removes *all files* in your current directory. (You had better be *very* sure that's what you wanted to say!)

[illegible]

5. [If italics were used for second emphasis]

> construct $A(I,J)$ is assumed to be a function call if no declaration for

[illegible]

NOTE that the Transcriber's Option Symbols also have meaning in the literary and Textbook Format codes - "their" in literary, and the boldface type indicator in textbook. This should cause no problems in displayed notation where all transcription is character by character. However, if you are going to use the textbook boldface indicator in your primary code notation, you should not use that Transcriber's Option Symbol for computer notation, to avoid any conflict.

XII TERMINATION INDICATOR
(Section XVI - CBC)

The rules concerning the use of the two-cell symbol - dots 4-5-6, 1-5-6 - as a Termination Indicator are distinct from those concerning the use of the symbol for an Ending Indicator.

As a Termination Indicator this symbol is always used to signal the end of embedded computer notation and the return to the primary code. A Termination Indicator is never used in displayed computer notation.

Rules and examples for the use of the Termination Indicator have been included in each of the previous indicator sections.

Because there may be some confusion regarding its use, a summary of the rules for the 4-5-6, 1-5-6 symbol, both as an End Indicator and as a Termination Indicator, is included here in Section XIII.

XIII SUMMARY OF THE RULES FOR THE USE OF THE END AND TERMINATION INDICATOR

⠸⠱

The two-cell symbol, dots 4-5-6, 1-5-6, has a multipurpose use. In order to avoid confusion, it should always be referred to by its proper name (i.e., End Nemeth Indicator, End Shape Indicator, End Half-Shift Indicator, and Termination Indicator).

(1) Nemeth Code Indicator - An End Nemeth Code Indicator is required to end computer notation that has begun with the Begin Nemeth Code Indicator.

a. '01001' ∉ L,

⠄⠴⠁⠴⠴⠁⠄ ⠸⠩⠄⠁⠑⠸⠱ ⠸⠸ ⠄

(2) Shape Indicator - An End Shape Indicator is required to end computer notation that has begun with the Begin Shape Indicator.

a.

⠸⠩⠅⠗⠑⠎⠑⠞⠸⠱

b.

For example:

123456

indicates that the kursor is

(3) Half-Line Shift Up and Half-Line Shift Down Indicators - An End Half-Line Shift Indicator is required to end computer notation that has begun with either of the Half-Line Shift Indicators

(a) if this notation is followed, without a space, by addition computer notation.
(b) if this notation is followed by text in the primary code.

a.

/KEYWORD!string$_1$!string$_2$

b.

(4) Transcriber's Option Symbol - When a Transcriber's Option Symbol has been designated as representing a particular type style, the two-cell symbol, dots 4-5-6, 1-5-6, is assigned as an End Transcriber's Option Indicator and is required to end computer notation that has begun with a Transcriber's Option Symbol.

a. [Assume dot matrix type style is being represented by the Transcriber's Option.] Displayed notation

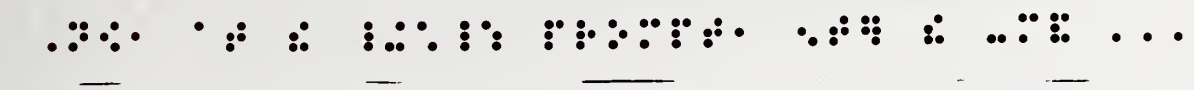
```
?SYNTAX ERROR
```

b. [Assume italics is being represented by Transcriber's Option.] Displayed notation

```
write(top);
write(bottom);
writeln
```

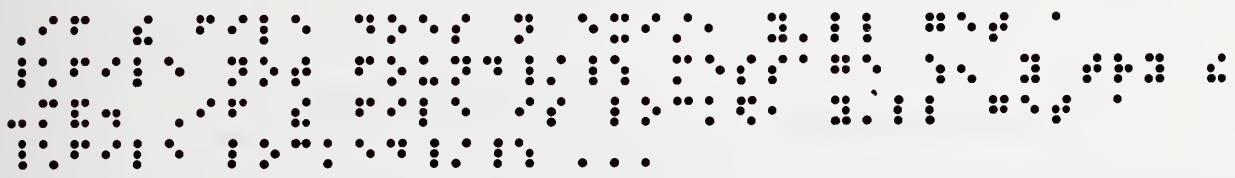
(5) Computer Braille Code Indicator - A Termination Indicator is required to end computer notation that has begun with the Begin Computer Braille Code Indicator, to signal a return to the primary code. (Technically speaking, the End Computer Code Indicator and the Termination Indicator are one and the same.)

a. Now, at the * prompt, enter the command



(6) Emphasis Indicator - When the notation is embedded, a Termination Indicator is required to end computer notation that has begun with the Begin Emphasis Indicator, to signal a return to the primary code. The Termination Indicator is in addition to the End Emphasis Indicator.

a. If the file does not exist, you'll get a FILE NOT FOUND message when you try the command. If the file is locked, you'll get a FILE LOCKED



b. echo file *filename* does not exist

(7) Caps Lock Indicator - When the notation is embedded, a Termination Indicator is required to end computer notation that has begun with the Caps Lock Indicator, to signal a return to the primary code. (Technically speaking, there is no End Caps Lock Indicator. The Caps Release Indicator, while it does end the effect of the Caps Lock Indicator, is used only for the purpose of signaling the occurrence of following lower-case letters, and is never used to signal the return to the primary code.)

a. Included in the catalog is the file DELETE.ME.1.

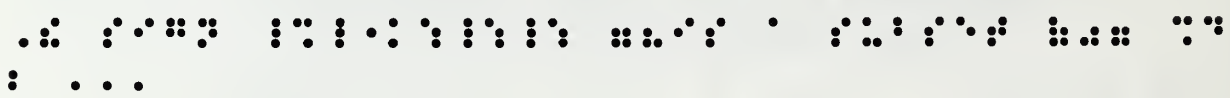
b. One implementation is to loop over the directories named in PATH, search-

(8) When the notation is embedded and either of the items in (1)-(2) is the only expression of computer notation, a Termination Indicator is required to signal the return to the primary code, in addition to the End Indicator.

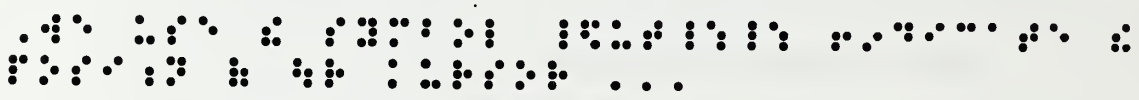
a.

42. a. Express $\frac{1}{9}$, $\frac{5}{9}$, and $\frac{7}{9}$ as repeating decimals.

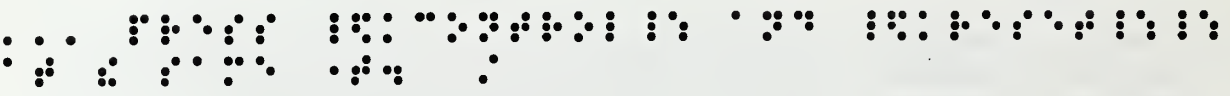
b. The sign $\subseteq$ ("is a subset of") should be



c. We use the symbol ▲ to indicate the position of our kursor

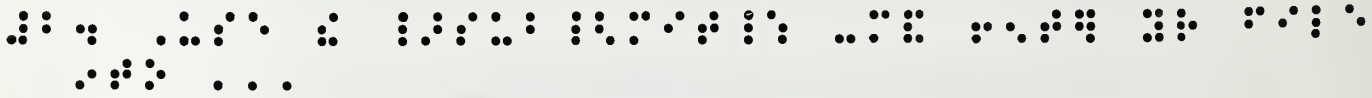


d. press CONTROL and RESET at the same time.



(9) When the notation is embedded, computer notation that has begun with any of the indicators in (5)-(7), and which contains any combination of these indicators, requires only one Termination Indicator to signal a return to the primary code.

a. 2. Use the SUBmit command to enter your file into



b.

[PAC[K] RET[RIEVE] file[. . ..file] [FROM masterfile] retrieves the specified file(s) from a masterfile and places the file(s) in your disk space. A duplicate of the file(s) also remains in the

[illegible]

Assume the following examples are embedded.

c. v$ = INPUT$(x[,[#]file number])

[illegible]

d. [Italics are emphasized notation.]

FOR *variable* = x TO y [STEP z]

[illegible]

e. [Embedded]

$ cat which # which cmd: which cmd in PATH is executed, version 1

[illegible]

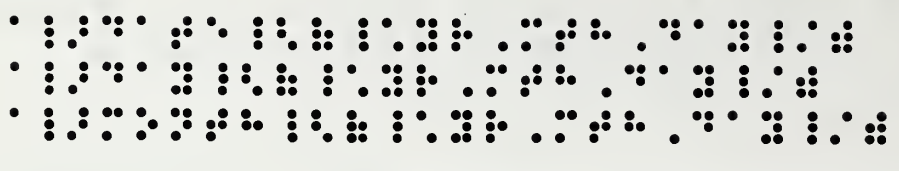
f. [Embedded]

@Dxxxx(*inp_range,col_offset,crit_range*)

[illegible]

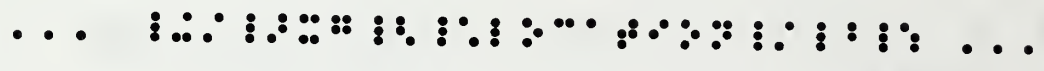
g. [Displayed; separate examples]

@DATE(*yr,mth,day*)
@DAY(*yr,mth,day*)
@MONTH(*yr,mth,day*)



h. [Embedded]

/XG*location*–

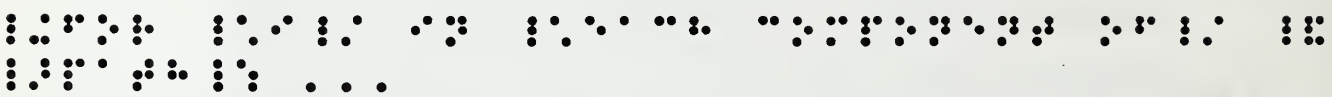


i. [Embedded]

v=ERDEV

j. [A "program" line inserted in text]

for *i* in *each component of* PATH



(10) When the notation is embedded, computer notation that has begun with any of the indicators, and which ends with any of the indicators in (1)-(4), requires a Termination Indicator to signal a return to the primary code, in addition to the End Indicator. NOTE: This is the "nesting" concept as explained in Section XVI of the CBC. Although it is theoretically possible for more than two of the 4-5-6, 1-5-6 symbols to appear together at the end of computer notation, the writer could not find any examples for this. (An example was devised in Section X to show this construction.)

Assume the following examples are embedded.

a. [Italicized type is second emphasis.]

do *if given name is in directory i*

b.

/KEYWORD.string$_1$.string$_2$

[illegible]

c. [Period is the end of the sentence.]

- ('01$^+$')$^+$.

[illegible]

d.

If necessary use CTRL + F3 and CTRL + F4 to block text

[illegible]

e.

To retrieve the file, use the command

get sandy,d2 **Return**

[illegible]

(11) When the notation is embedded, computer notation that has begun with any indicator, and that contains any indicator, but that ends with notation under the effect of any of the indicators in (5)-(7), requires only one Termination Indicator to signal a return to the primary code.

a.

The shell stores the exit status of the last program in the variable `if` *command* `$?`:

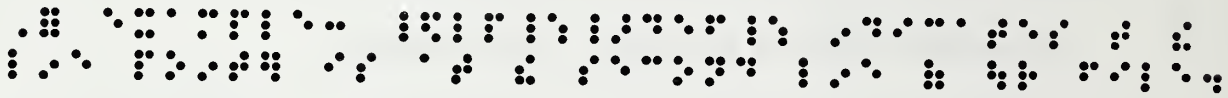
[illegible]

b.

To change the name of PHONE LIST (fn1) to DIRECTORY (fn2),
type RENAME SPACE PHONE SPACE LIST, DIRECTORY

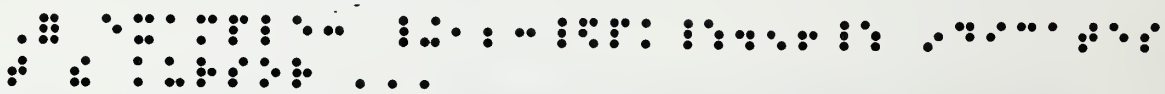
c.

For example: ►DEF indicates that the line pointer is at the second line of our file.



d.

For example: 123▲456 indicates that the kursor is between the third and fourth characters on the current line.



Assume the following examples are embedded.

e. `$ mail mary` *Send mail to* `mary`

XIV COUNTABLE SPACES INDICATOR
(Section IX - CBC)

Add these rules:

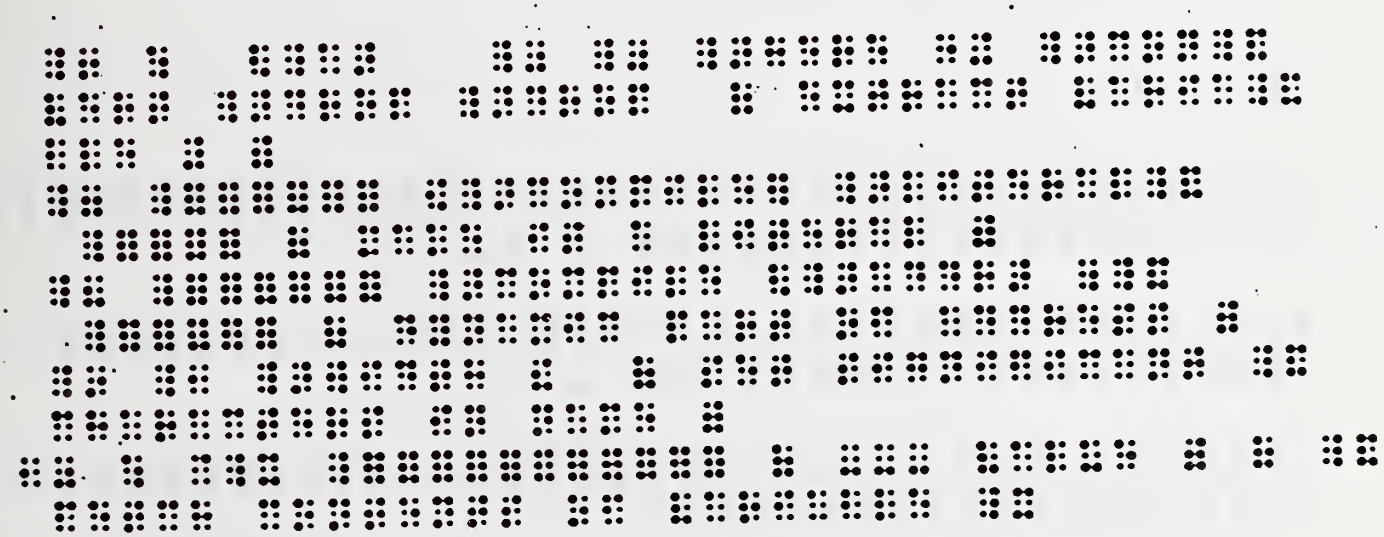
When a series of spaces must occur at the end of a line which has a runover, at least 2 full cells of the countable spaces indicator must appear at the end of the line.

SUGGESTION: Although the countable spaces indicator is not necessary for leading spaces in programs where the CBC indentation rules apply, there are some programs which require vertical spacing. Occasionally this vertical spacing requires more than 40 spaces at the beginning of a line. When this situation occurs, use the Countable Spaces Indicator for the leading spaces. (See Example 2 below.)

NOTE: In Example 9-2, CBC, a broken bar is transcribed as a vertical bar. The broken bar is usually the print result of a particular printer and does not have a particular computer meaning. However, if a text uses both a broken and unbroken bar, or if the broken bar is described as such in the text, use the shape indicator and assign a letter symbol to transcribe this symbol.

1. Countable Spaces Beginning on Runover Line

```
 6 : <A>   0 0 HERE 1+ CONVERT DROP DROP  ( extract variable # )
 7         [COMPILE] LITERAL              ( make it a literal )
 8         COMPILE <ADDR> ;               ( dynamic part of address )
 9 1 WIDTH !   ( set significant characters in name )
10 : @X              ( --- value ) ( fetch contents of variable X
```



2. LINE FEEDS

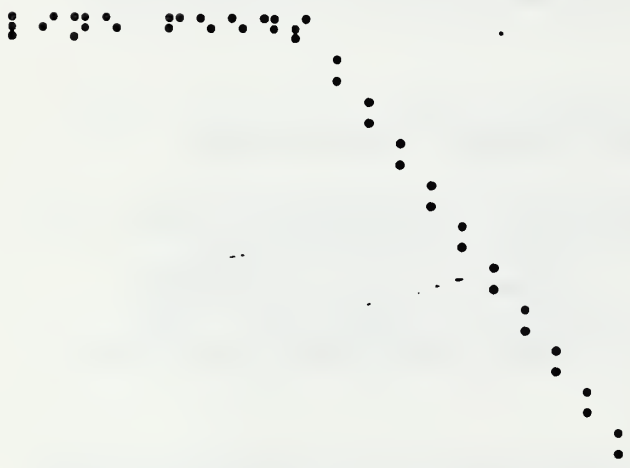
SPACING

! 60 SPACES

!

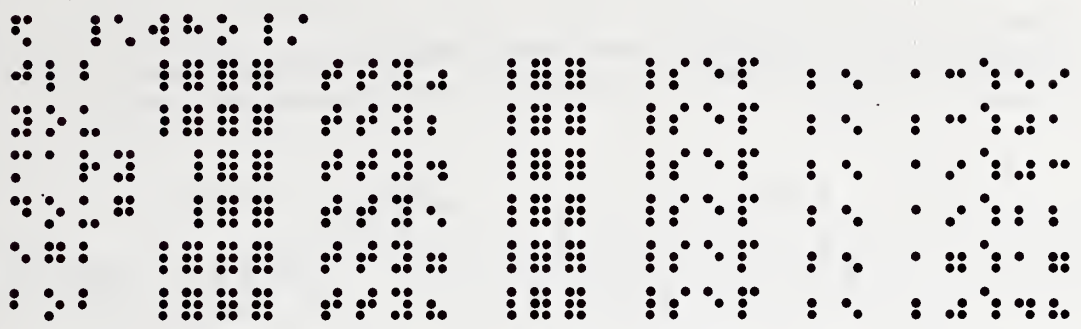
!

CHARACTERS
1234567890 1234567890
ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz abcdefghijklmnopqrstuvwxyz (lower case)
!"#$%&'()*+,-./ !"#$%&'()*+,-./



SUGGESTION: If countable spaces cannot be determined precisely, leave 5 spaces. Use the countable space indicator to fill these spaces.

```
$ who
jlb      tty0    Sep 25 13:59
you      tty2    Sep 25 23:01
mary     tty4    Sep 25 19:03
doug     tty5    Sep 25 19:22
egb      tty7    Sep 25 17:17
bob      tty8    Sep 25 20:48
```



XV CONTINUATION INDICATOR
(Section VI - CBC)

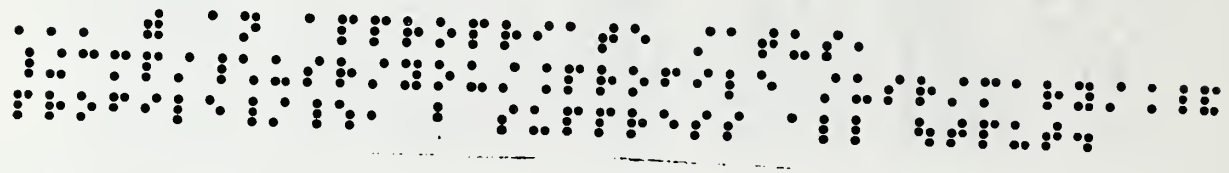
In displayed computer notation, if the print line doesn't fit on the braille line, use the Continuation Indicator.

In embedded computer notation, if the print expression doesn't fit on the balance of the braille line:

a. drop the expression down to the next line.
b. if the expression still will not fit, use the Continuation Indicator.

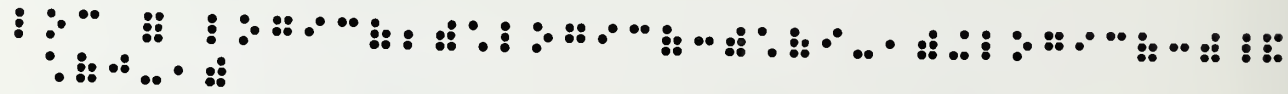
1. [Embedded]

A few commands, such as cmp and grep, have an option -s that causes them to exit with an appropriate status, *cmp /usr/you/.profile /usr/mary/.profile*, but suppress all output.



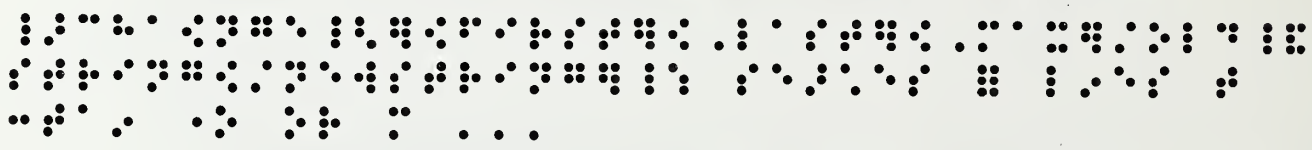
2. [Displayed]

```
LOC = LOGIC(2)*LOGIC(3)*(I-1)+LOGIC(3)*(J-1)
```



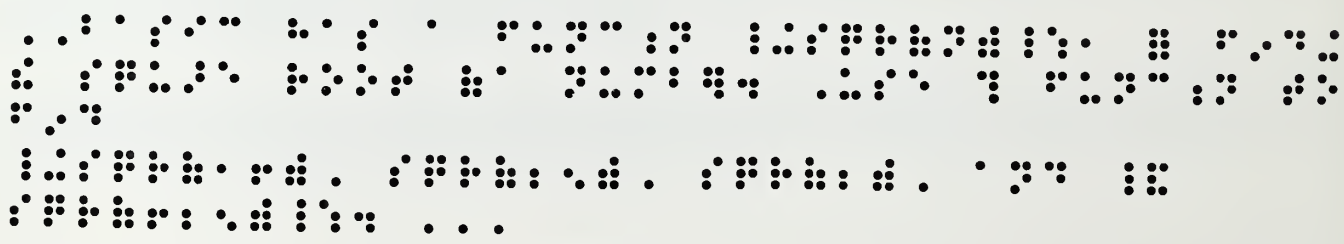
3.

CHA[NGE][first][,last][,max]/oldstring[/newstring] searches for lines that contain one or more



4.

BASIC has a function, SQR(N), for finding the square root of a number. Use this function to find SQR(16), SQR(25), SQR(2), and SQR(625).



SUGGESTION: Divide a line at any logical place. If a word is involved, divide after a syllable if such a division will save having to make a further runover line. Divide at a space otherwise. Be sure to put the space before the Continuation Indicator.

XVI ISOLATED LOWER-CELL SIGN
(Section VIII - CBC)

Section VIII in the Computer Braille Code is self-explanatory.

Here are a few more examples of the use of this sign.

1. "RUN 1 of 3".

2. 1 2 - 4 5 + * =

Parentheses are not needed.

3. ... namely +, - and * (or PLUS, MINUS and TIMES).

XVII REPRESENTATION OF CONTROL CHARACTERS
(Section V - CBC)

The following examples are all transcribed with the shape indicator. (See Section VII.)

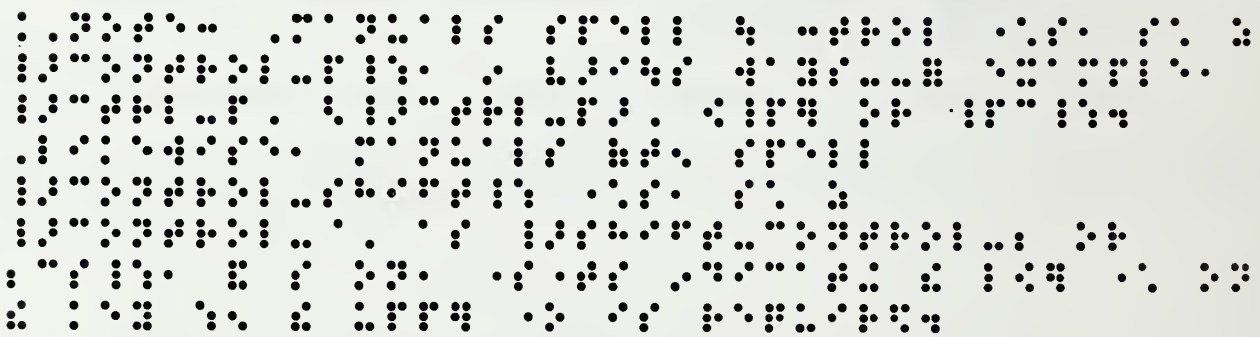
[Braille examples are in shape section.]

Enter. Begins or continues the specified function.

Tab. Moves the cursor forward to the next item on the menu or form. With Shift, Tab moves to the previous item.

Backspace. Moves the cursor to the left one space and erases the character in that location.

Note: Manuals spell out control characters, such as CONTROL-P, in various ways—for example, CTRL-P, (CTRL-P), [P] or Pc. Likewise, manuals often spell CONTROL-SHIFT characters, such as CONTROL-@, as SHIFT-CONTROL-2 or 2cs, and so on, sometimes indicating the lower character on the key when the upper one is required.

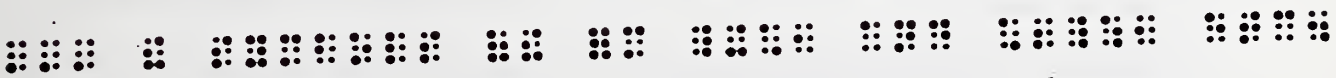


XVIII SUMMARY OF RULES FOR COMPUTER CODE NOTATION

I. General Rules
   A. Transcribe character for character - no contractions
   B. Do not use the letter sign or capital sign with letters.
   C. Transcribe numbers in the lower two-thirds of the cell - do not use the number sign.
   D. Do not use the contractions for to, into or by preceding computer notation.
   E. Frequent switches between computer notation and the primary code should be avoided, particularly on the same braille line. This includes
      Table columns
      Titles, headings, or sub-headings
      Expressions of computer notation separated by commas, semicolons, etc.
      Expressions of computer notation separated by one, or a few, words
      This rule is rather sweeping and may cause the transcriber some concern. You must rely on your experience for judging when to implement this rule. Generally, try to strike a balance between excessive indicators and excessive spelled-out words. One word between computer expressions could virtually always be included as part of the computer notation. Two or three words could be included, especially if those words do not contain contractions. Finally, use your good judgment in the use of this rule. If you are very unsure of whether to use it or not, don't.

... the symbols of the form *1, *2, etc.

... the symbols of the form *1 and *2, etc.



II. Embedded Computer Notation
   A. Embedded computer notation must begin with one of the following indicators.
      1. Computer Braille Code Indicator
      2. Caps Lock Indicator
      3. Emphasis Indicator
      4. Shape Indicator
      5. Nemeth Code Indicator

      No other indicator or computer symbol may initiate computer notation.
   B. Use the Computer Braille Code Indicator to initiate computer notation that begins with:
      1. Lower-case letter or letters
      2. A single upper-case letter - the Shift Indicator must also precede this upper-case letter. (See Section IV for the definition of a single upper-case letter.)
      3. A number
      4. A computer symbol
      5. A superscript or subscript (see Section IX)
      6. A Transcriber's Option Symbol (see Section XI)

When the Caps Lock, Emphasis, or Shape Indicator is used to initiate computer notation, all symbols and rules of the Computer Braille Code are in effect and may be used without being indicated by the Computer Braille Code Indicator.

C. Use the Caps Lock Indicator to initiate computer notation that begins with two or more upper-case letters. The two upper-case letters may be separated by other computer symbols. If this notation is followed, without a space, by lower-case letters, use the Caps Release Indicator immediately following the last upper-case letter. (see Section V.) (Also see IV below.)

D. Use the Emphasis Indicator to initiate computer notation that begins with emphasized material. Use the End Emphasis Indicator when this material ends. (see Section VI.)

E. Use the Shape Indicator to initiate computer notation that begins with a shape which is not represented in the Computer Braille Code. Use the End Shape Indicator when this notation ends. (see Section VII.)

F. Use the Nemeth Code Indicator to initiate mathematical expressions, or when no computer braille code symbol exists. Use the End Nemeth Code Indicator when this notation ends. (see Section VIII.)

G. Use any of the indicators in C, D, E, or F above when needed within computer notation.

H. Use the Isolated Lower-Cell Sign when needed. (see Section XVI.)

I. Use the Continuation Indicator when computer notation must be run over to a new braille line. (see Section XV.)

J. Use the Countable Spaces Indicator if a series of five or more spaces must be indicated within computer notation. (see Section XIV.)

K. A Termination Indicator must be used to end embedded notation and signal the return to the primary code.

III. Displayed Computer Notation

A. The parts of this section already noted that apply to displayed computer notation are:
   1. Sec. I, A, B, C
   2. Sec. II, C, D, E, F, G, H, I, J

B. The Computer Braille Indicator is not used in displayed computer notation.

C. A Termination Indicator is not necessary to signal a return to the primary code at the end of displayed computer notation. (Do not confuse the dots 4-5-6, 1-5-6 "ending indicator" with the dots 4-5-6, 1-5-6 "Termination Indicator".)

IV. When computer notation is predominantly upper-case, but there are instances when notation contains lower-case letters, reverse the meaning of the Caps Lock and Shift Indicators. Include a transcriber's note explaining this procedure.

Use the Shift Indicator for single lower-case letters.
Use the Caps Lock Indicator for two or more lower-case letters.
Use the Caps Release when a string of lower-case letters is followed, without a space, by upper-case letter(s).
Use the Termination Indicator to signal a return to the primary code.

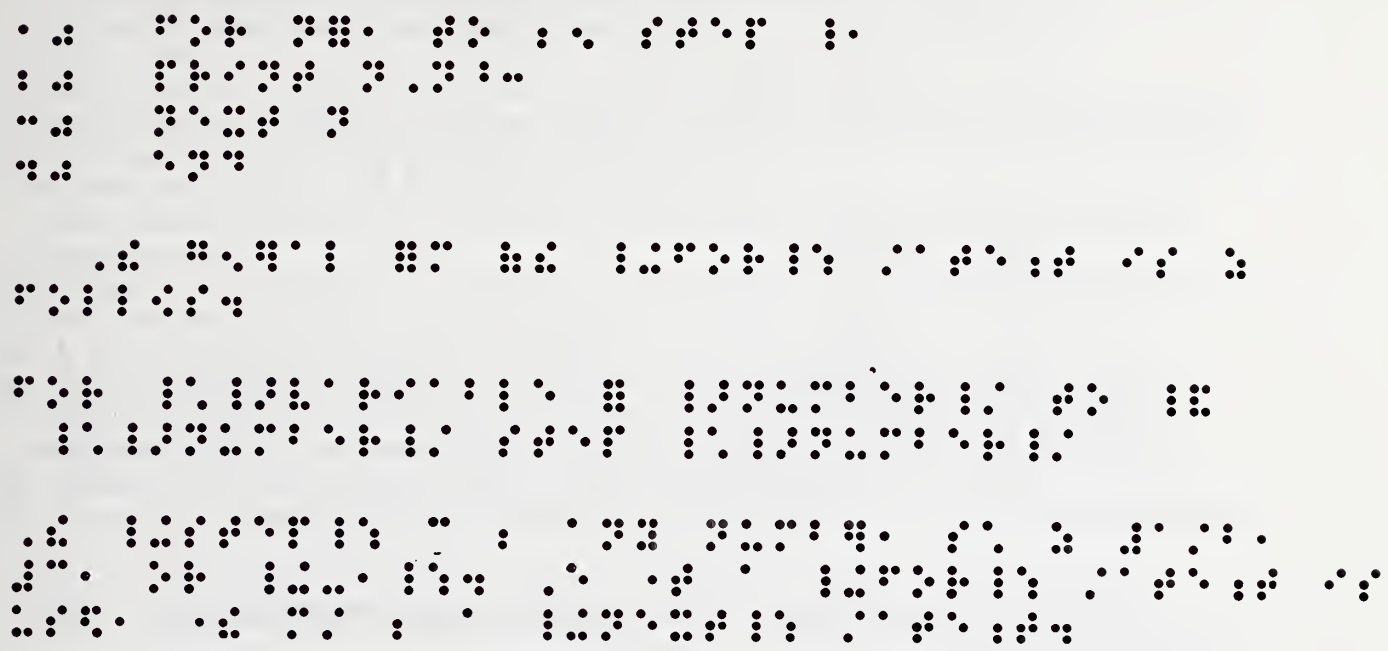
1.

```
10 FOR N=1 TO 25 STEP 1
20 PRINT N,N↑3
30 NEXT N
40 END
```

The general form of the FOR statement is as follows.

FOR *variable* = *number* TO *number* STEP *number*

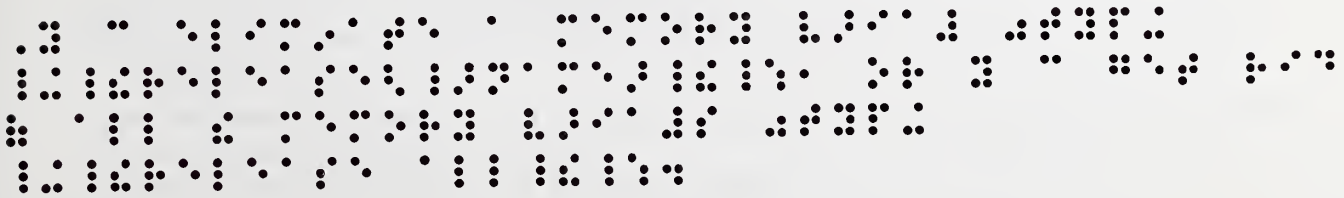
The STEP can be any number, such as 1/2, 3, or −1. Every time a FOR statement is used, there must be a NEXT statement.



2. [Text is predominantly upper-case for computer notation. Transcriber's Option has been assigned to the diamond symbol.]

```
VARIABLE
ANOTHER
THIRD
CONSTANT
**TOTAL**
```

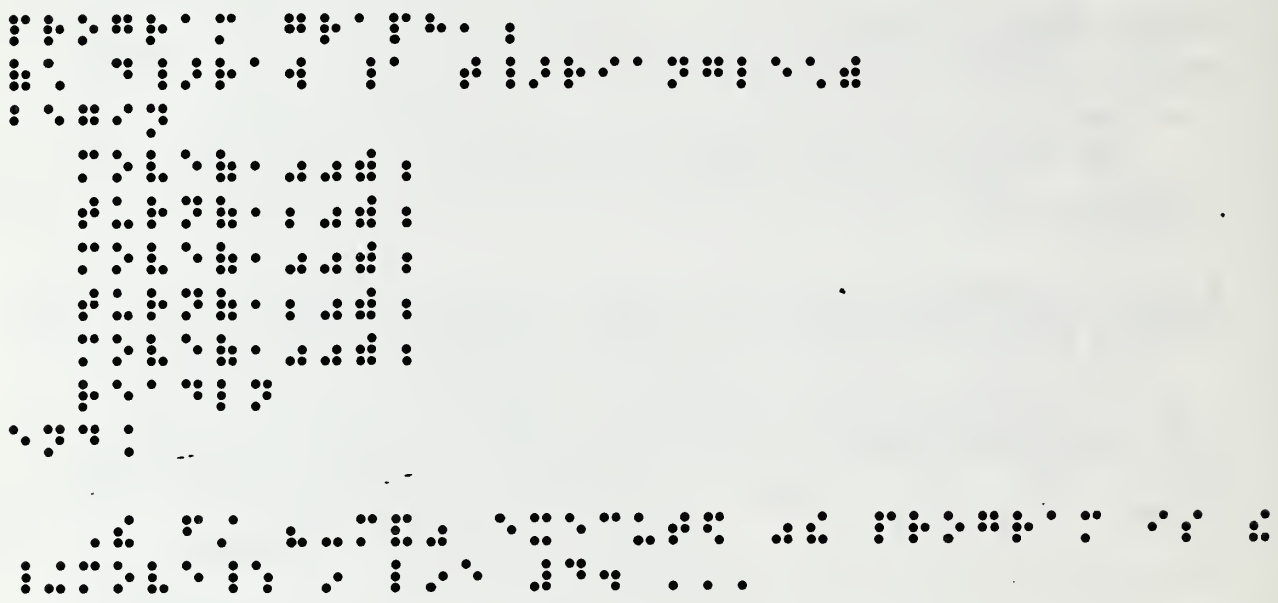
You can eliminate a memory variable by typing ♦RELEASE<name>♦, or you can get rid of all the memory variables by typing ♦RELEASE ALL♦.



3. [Computer notation is predominantly upper-case. Only occasionally is there a comment in lower-case.]

```
PROGRAM GRAPH1;
(* Draw a Triangle*)
BEGIN
  MOVE(100);
  TURN(120);
  MOVE(100);
  TURN(120);
  MOVE(100);
  READLN
END.
```

The first "command" executed by this program is the MOVE in line 4. Commands in lines 5, 6, 7, 8, and 9, are then executed in that order. When



V. When computer notation is exclusively upper-case, Case Indicators are not necessary. (see Section V.)

XIX SUMMARY OF FORMAT
(Section III - CBC)

1. Use 41 cells if transcribing on a braillewriter.
2. Use 40 cells if transcribing on a computer.
3. List codes used on Transcriber's Notes page.
4. Devised symbols or those assigned special usage should be listed on the Special Symbols page. SUGGESTION: List Computer Braille Code Symbols (not letters or numbers) and indicators to be included in each volume until the Computer Braille Code is available in braille.)
5. A blank line should precede and follow displayed computer notation. If displayed material is not entirely computer notation, use the rules for format of the primary code for this material.
6. Begin displayed computer notation in cell 1 with runovers in cell 2. NOTE: Some programs will need to follow print spacing.
7. Transition to a new braille page takes the place of the blank line preceding or following displayed computer notation.
8. Whenever possible, an entire program should be transcribed on a single page. (See Section XX for complete rules regarding the transcription of programs.)
9. Unspaced expressions of computer notation should never be divided between lines (should never be hyphenated).
10. A program line should not be divided between pages.
11. Begin embedded computer notation on a new braille line if it will not fit on the current line. If it will still not fit, use the continuation indicator and continue on the next line. Text may then resume on the same line.
12. Runovers of embedded notation start at the established margin.
13. Format for labeled figures or diagrams should be that of the code in effect.

XX DISPLAYED COMPUTER NOTATION AND PROGRAMS - FORMAT

All displayed computer notation is not a program or program line(s). However, the format is the same and references to programs here include all forms of this displayed notation. (For displayed notation that is not a "program", see Section II)

A. Unlabeled programs:

A blank line must be left before and after the program.

Whenever possible, the program must be transcribed on a single braille page.

When the program is too long to fit on the current page following text, start it on the next braille page. (The balance of the current page will be left blank.) Transition to the new page will take the place of the required blank line.

Do not put a program line on line 1 if no running head is used, or on line 25 unless more than 5 spaces can be left between the end of the program line and the page numbers.

See D. below for the body of the program.

When the program takes more than one braille page, use the format for labeled programs.

Text following a program that takes more than one page must be started on a new braille page.

Do not divide a program line between pages.

Text following a program that takes more than one page must be started on a new braille page.

Do not divide a program line between pages.

B. Labeled Programs

A labeled program must begin on a new braille page. (Exception: see C. below.) No text may follow on the same page.

When a reference to the labeled program is included in the text, finish the braille page containing text and then insert the program on the page following the reference.

When no reference to the program is included in the text, insert the program at the most likely spot after the current braille page has been completed.

If the program is on a print page different from the one that was transcribed at the time the figure was referred to, the print page number must be included within parentheses following the identifying label. The print page numbering should follow the sequence in effect on the braille page preceding the program.

Begin the identifying label number or title in cell 1 on line 1 (line 2 if you are using a running head). If the label contains more than just a title or a number, continue on the same line with the rest of the label in paragraph form.

Leave a blank line after the complete label.

Transcribe the program (see D. below.).

Center the identifying label on line 25. Line 24 should be left blank.

Continue the text on the next braille page.

When a labeled program takes more than one braille page, start the first program line on line 2 of the braille page, whether or not a running head is used.

C. When all programs, and all or most program segments, are labeled (i.e., with figure numbers) and the programs are often very short, and they appear in the continuity of the text, follow the print as to the placement of the programs.

1. Leave a blank line before the programs.
2. Center the label and leave a blank line after it.
3. Transcribe the program lines.
4. Leave a blank line after the program.
5. Continue with the text.
6. As with any program, whenever possible it should be transcribed on one braille page. In the case where one of these programs will not fit on the current braille page following text, the balance of this braille page must be left blank. Begin the program at the top of the next page. Text may then follow on the same braille page.
7. When one of these programs takes more than one braille page, use the instructions in A.2. a-c above.

D. Transcribing the Body of the Program

Ascertain the lines beginning at the leftmost margin. All other lines should be related to the leftmost line(s) as to placement. (The first line of a program is not always the leftmost line.)

When lines within a program are indented, braille lines should also be indented. Each braille level of indentation should begin two spaces to the right of the previous level, regardless of the spacing used in print. Successive levels of braille indentation begin in cells 1, 3, 5, etc. (Also see Section XXII.)

A program line cannot be divided between pages.

A continuation indicator should be used when the entire print line will not fit on the braille line. Runovers should always begin in cell 2, regardless of the current level of indentation.

E. When arrows, enlarged braces, enlarged brackets, etc. are used to set off or point out lines in the program, or to show explanatory notes, draw the arrows, brackets, etc., or explain their placement and function in a transcriber's note.

1.

```
IFNONE
IFANY
IFALL   } (t1)(t2)...(tn)  THEN  (op1)(op2)...(opm)  LABEL
IFNALL
```

2.

```
C SAMPLE PROGRAM 3-13
C TREATMENT OF STUDENTS AS 10 CLASSES OF N STUDENTS EACH
      READ, J
      FJ = J
      DO 8 I = 1, 10
      SUM = 0.
      READ, N
      DO 4 K = 1, N
      READ, NWRONG
      GRADE = (FJ - NWRONG)/FJ * 100.
      SUM = SUM + GRADE
4     PRINT, I, K, GRADE
      GRADE = SUM/N
8     PRINT, I, GRADE
      STOP
      END
```

3.

Syntax	Examples
COMMAND fn[.Sn][.Dn][.Vn] (Arguments)	SAVE ELEGANT CODE.D2 DELETE BOOKS,S6.D1,V3 RUN WHIZBOOM.D1

Command — Filename — Slot Number — Drive Number — Volume Number

F. "Comments" in programs take many forms. Some comments are a programming device and are transcribed using the format of the CBC. Some are the author's explanations, a teaching device, and will need a special format that can be worked out by the transcriber. The text usually - but not always- describes any special symbols, type style, or format that is used for comments. A thorough examination of the text should be made in order to determine the proper way to transcribe comments.

See Section XXI for some suggestions and examples for comments.

G. Program lines are often listed with a number preceding each line. In most cases these numbers are a multiple of 5 or 10 and are an integral part of the program itself. However, in some instances program lines are numbered by the author as a reference for the explanation of the program lines. (This device is always explained in the text.) This numbering always starts with #1 and continues consecutively throughout the program. If the program were listed, these numbers would not appear. (Just as when an author's comments are included in the text - if the program were listed, the author's comments would not appear.)

It is suggested that these numbers be transcribed outside the program. Use the primary code to transcribe the numbers, starting in cell 1. If the numbers have periods or colons following, transcribe the punctuation also. Leave spaces after the numbers so that there will be two spaces after the longest number. (In a listing that has 2-digit numbers, leave three spaces after the 1-digit numbers. If the listing has 3-digit numbers, leave four spaces after the 1-digit numbers and three spaces after the 2-digit numbers.) After the spaces, start program lines as usual. Indent program lines as specified in the CBC. Do not use the Countable Space Indicator for spaces between the reference numbers and the beginning of a program line. These spaces are to be construed as leading spaces before a program line. Runovers of program lines should start one space to the right of the left-most program line. In other words, the program should be transcribed as if the reference numbers were not there.

Here are some examples of programs with reference numbers.

1. [Primary computer notation is regular type - boldface type is assigned to the emphasis indicator.]

The following explanation of the details of the program structure is keyed to the line numbers on the left in Fig. 11-1, which are not a part of the ALGOL program.

```
Line
 1        begin
 2           procedure SUM(V,N); value N;
 3              real array V; integer N;
 4              begin integer I; real TEMP;
 5                 TEMP:=0;
 6                 for I:=1 step 1 until N do TEMP:=TEMP+V[I];
 7                 SUM:=TEMP
 8              end;
 9           integer K;
10        START: inreal(1,K);
11                 If K > 0 then
12                    begin real array A [1:K];
13                       inarray(1,A);
14                       outreal(2, SUM(A,K));
15                       goto START
16                    end
17        end
```

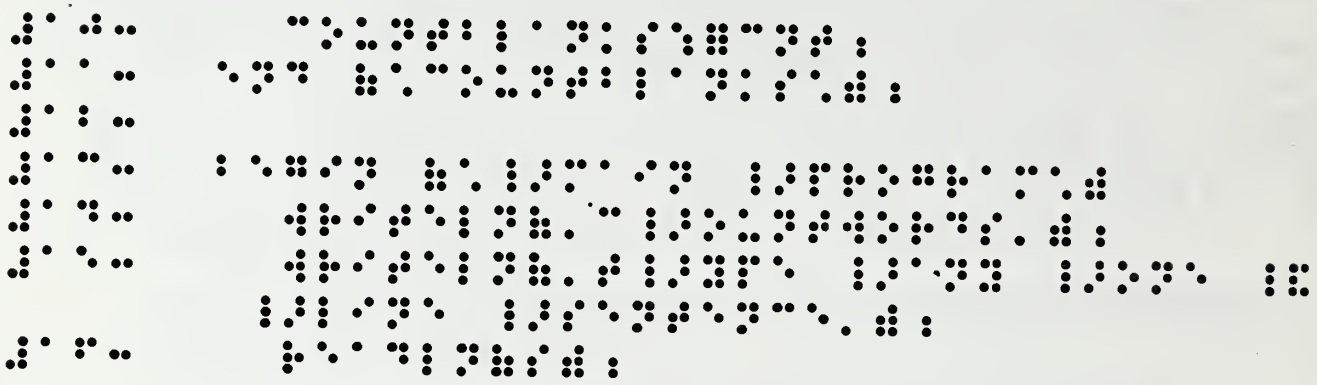
Fig. 11-1. Example ALGOL program.

2. [Computer notation is exclusively upper-case.]

```
Line
   1        TEST: PROCEDURE OPTIONS (MAIN);
   2              START:  GET LIST (K);
   3                      IF K > 0 THEN BEGIN;
   4                           DECLARE  A(K) DECIMAL FLOAT;
   5                           GET LIST (A);
   6                           PUT LIST ('INPUT IS',A,'SUM IS',SUM(A));
   7                           GO TO START;
   8                         END;
   9              SUM: PROCEDURE (V);
  10                      DECLARE V(*) DECIMAL FLOAT,
```

3. [Computer notation is predominantly upper-case.]

```
 1: PROGRAM COUNTWORDS;
 2: VAR S:STRING;
 3:
 4: FUNCTION COUNTBLANKS(S:STRING):INTEGER;
 5: VAR CNT,K:INTEGER;
 6: BEGIN
 7:   CNT:=0;
 8:   FOR K:=1 TO LENGTH(S) DO
 9:      IF S[K]=' ' THEN CNT:=CNT+1;
10:   COUNTBLANKS:=CNT;
11: END (*COUNTBLANKS*);
12:
13: BEGIN (*main program*)
14:   WRITELN('Countwords');
15:   WRITELN('Type any one line sentence');
16:   READLN(S);
```

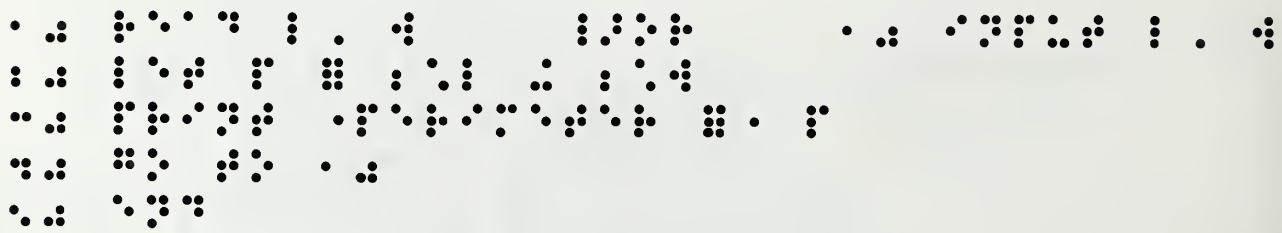


These are programs with numbered lines that are part of the program.

4.

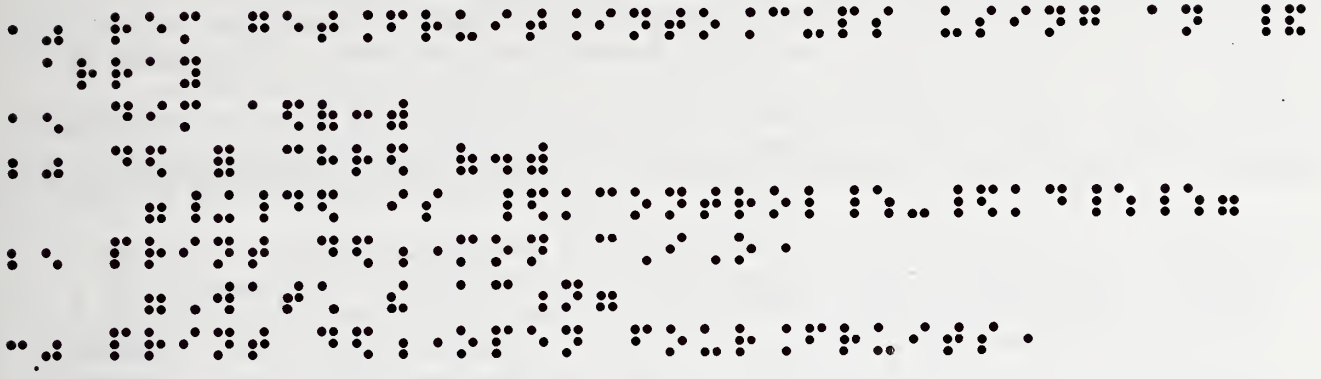
```
10 READ L, W
20 LET P = 2*L + 2*W
30 PRINT "PERIMETER =" P
40 GØ TØ 1C
50 END
```

```
10 INPUT L, W
```



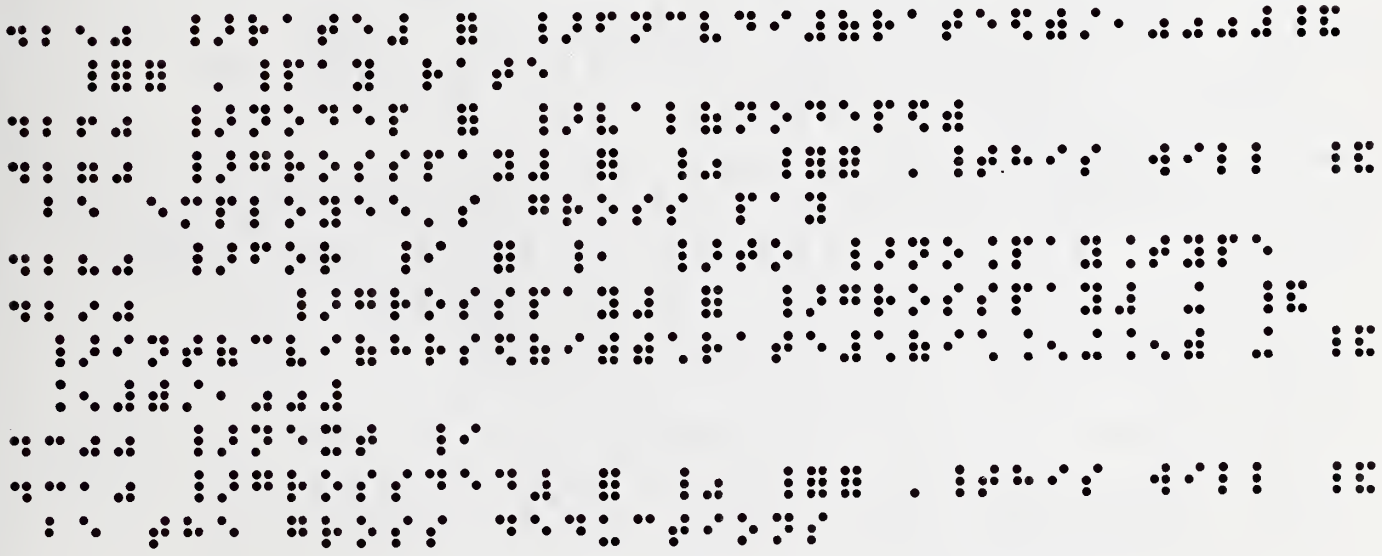
5. Format for author's comments; is explained in a Transcriber's Note.

```
10 REM GET.FRUIT.INTO.CUPS USING AN ARRAY
15 DIM A$(3)
20 D$ = CHR$ (4)          (D$ is CONTROL-D)
25 PRINT D$;"MON C,I,O"   (Watch the action)
30 PRINT D$;"OPEN FOUR.FRUITS"
```



6. [See "Comments" section for further examples of this program.]

```
4250 RATE# = FNCVDI#(RATE$)/10000      'Pay rate
4260 NODEP = VAL(NODEP$)
4270 GROSSPAY# = 0                     'This will be employee's gross pay
4280 FOR I = 1 TO NO.PAY.TYPE
4290    GROSSPAY# = GROSSPAY# + INT(CVI(HRS$(I))*RATE#*(I*.5+.5) + .5#)/100#
4300 NEXT I
4310 GROSSDED# = 0                     'This will be the gross deductions
```

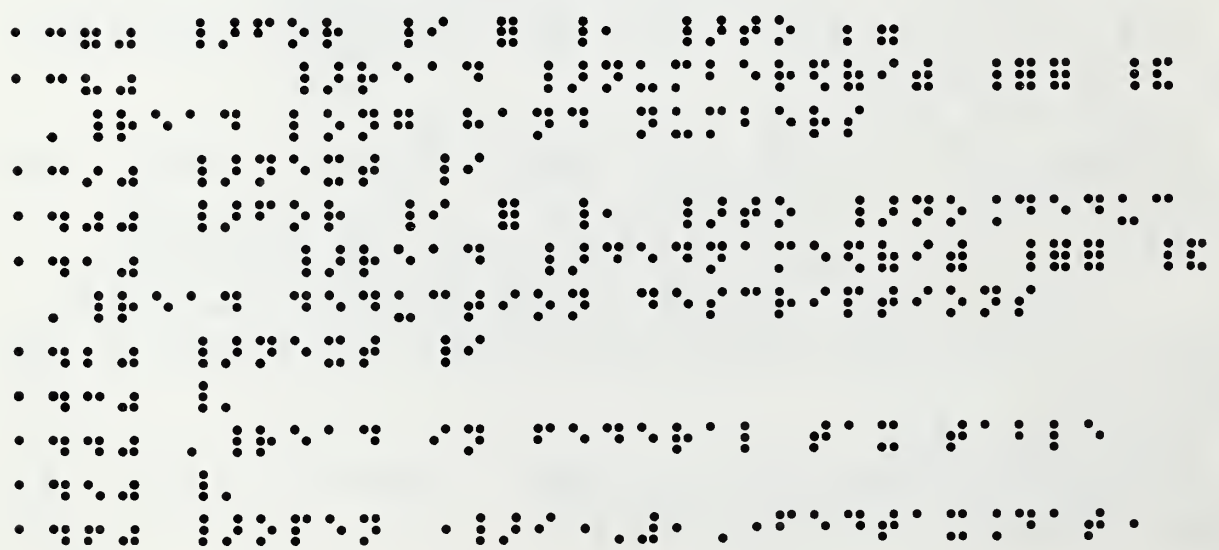


XXI COMMENTS

Virtually every program contains "comments". Comments which are entered into the program by the programmer to aid in reading and understanding the program are set off from the program lines themselves by some symbol (e.g., parentheses, brackets, quotes, etc.) or are identified in some other way (e.g., REM in BASIC) and are always in the same type style as the program lines (although they may be lower case letters when the program lines are all upper-case). Except for one or two programming languages, the placement of comments is arbitrary and is left up to the programmer. (REM in BASIC is an exception.) The symbols that are used to set comments off from program lines are always explained in the text. Even though the placement of this type of comment is arbitrary as far as the programmer is concerned, the transcriber should space such comments as they are spaced in print, using the Countable Spaces Indicator when the comment is placed five or more spaces from the end of the program line. All of the symbols and rules of the CBC should be used in transcribing program comments. A transcriber's note at the beginning of the work should explain this format. Here are some examples of the different ways comments are represented in print.

1. In this program, comments are preceded by a single quote sign. They are printed as a sentence would be printed - single capital letter to begin, followed by lower-case letters. Comments are placed arbitrarily throughout. Notice the comment "sign" (single quote) at the beginning of a line, followed by spaces. This sets off the comment for the program reader, but no blank lines will be left when the program is run, since the compiler ignores anything after the comment sign.

```
1370 FOR I = 1 TO 27
1380    READ NUMBER$(I)          'Read long hand numbers
1390 NEXT I
1400 FOR I = 1 TO NO.DEDUC
1410    READ DEDNAME$(I)         'Read deduction descriptions
1420 NEXT I
1430 '
1440 'Read in federal tax table
1450 '
1460 OPEN "I",#1,"FEDTAX.DAT"
```



2. Notice that this program uses a combination of upper- and lower-case letters in the program lines, as well as in the comments. This program requires the use of the Caps Lock Indicator and Shift Indicator for upper-case letters.

```
1470 IF NOT EOF(1) THEN 1510
1480    PRINT FNCP$(5,1);"WARNING: No federal tax table has been entered."
1490    PRINT "Press any key to return to menu. ";:A$ = INPUT$(1)
1500    GOTO 3200              'Exit program
1510 INPUT #1,FTSIZE,MAXDEP,EXPRLT$
1520 DIM FTAX#(FTSIZE,MAXDEP),INDEX#(FTSIZE)
```

3. In this program, comments are preceded by the crosshatch, or number, sign. Comments are printed in the same type style as program lines and are arbitrarily spaced.

```
        if test -f $i/$1        # use test -x if you can
        then
                echo $i/$1
                exit 0          # found it
        fi
done
exit 1          # not found
$
```

In these programs, comments are enclosed in /* preceding and */ following the comment, and are printed in the same type style as the program lines and are arbitrarily spaced.

4.

```
#include  <stdio.h>

main()    /* rudimentary desk calculator */
{
     double sum, v;

     sum = 0;
     while (scanf("%lf", &v) != EOF)
          printf("\t%.2f\n", sum += v);
}
```

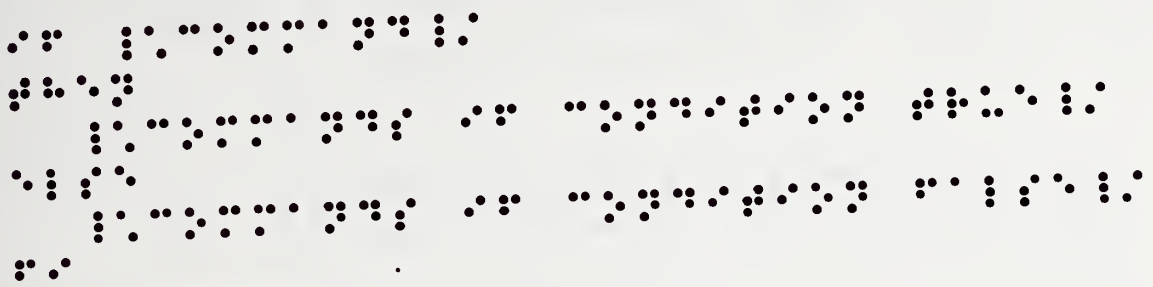
5.

```
#define MAXOP   20   /* max */
#define NUMBER  '0'  /* sign */
#define TOOBIG  '9'  /* signal that string is too big */

main()    /* reverse Polish desk calculator */
{
     int  type;
     char s[MAXOP];
     double op2, atof(), pop(), push();
```

In the programs in Examples 6 and 7 which follow, comments actually belong to the category of "comments by the author", which is covered here in the following pages. The comments are in italic type and are inserted in the program lines. Comments which are printed in this manner should be transcribed as they appear and the Emphasis Indicator should be used. (Or the Transcriber's Option symbol if more than one type style is being used for the program lines themselves.)

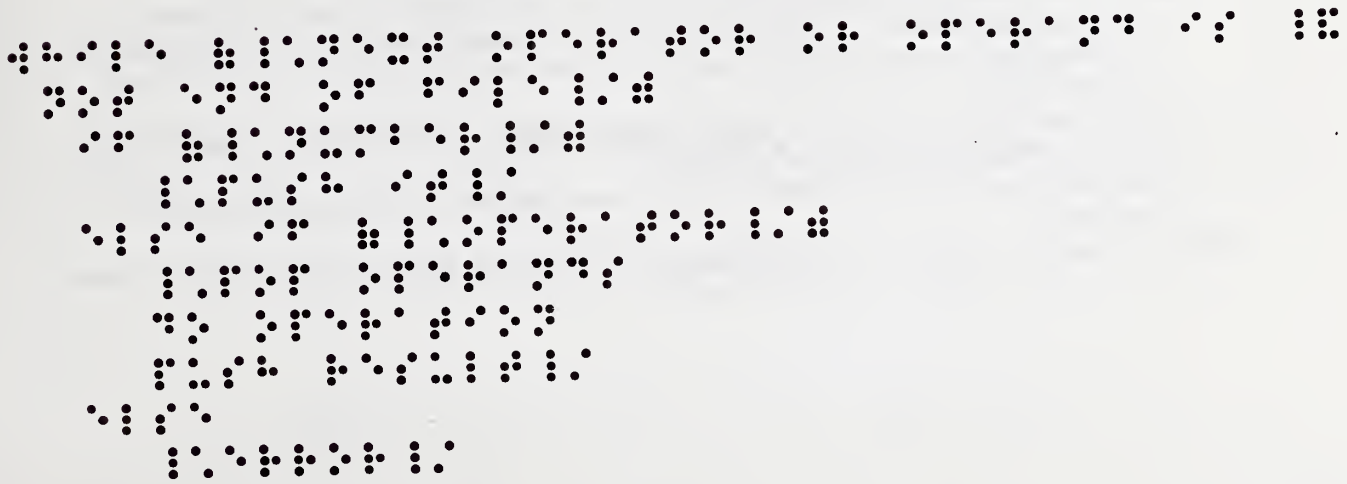
6.

```
if command
then
        commands if condition true
else
        commands if condition false
fi
```



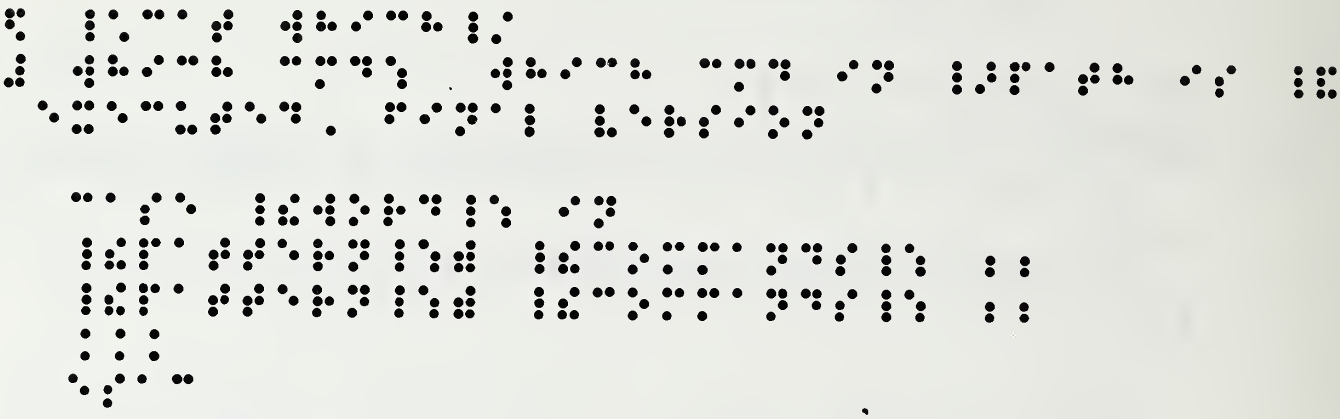
7.

```
while (next operator or operand is not end of file)
    if (number)
        push it
    else if (operator)
        pop operands
        do operation
        push result
    else
        error
```



8. Notice in this example that two type styles are used for program lines - slanted and regular boldface. The italic comments will require a Transcriber's Option symbol, as the Emphasis Indicator is used for slanted letters.

```
$ cat which
# which cmd:  which cmd in PATH is executed, final version

     case word in
     pattern)  commands ;;
     pattern)  commands ;;
     ...
     esac
```



Besides comments which are meant to be part of a program, most books concerning programs and programming languages contain comments which are added by the author to explain what the various program lines mean. These comments are almost always in a different type style from that of the program lines themselves, and are often set off in columnar form at the right side of the program lines.

The suggested method for transcribing this type of comment is:
 Drop down to the next line following the program line.
 Insert the comment starting five cells to the right of the beginning of the program line. (This will alert the reader to the fact that this is <u>not</u> another program line, which would have been two or four cells to the right.)
 Transcribe the comment in the <u>primary</u> code. Use italics, parentheses, etc. if they are used in print.
 Runovers should be blocked in the same cell as the beginning of the comment.
A transcriber's note at the beginning of the work should explain this format.

Here are some examples of this type of format.

9. [Author's comment is in italic type.]

```
$ ls -l test
-rwxrwxrwx 1 you          11 Oct  1 06:55 test     Still there
$ which which
/usr/you/bin/which
$ which test
./test
$ rm test
$ which test
/bin/test
$
```

10. [Author's comment is in italic type.]

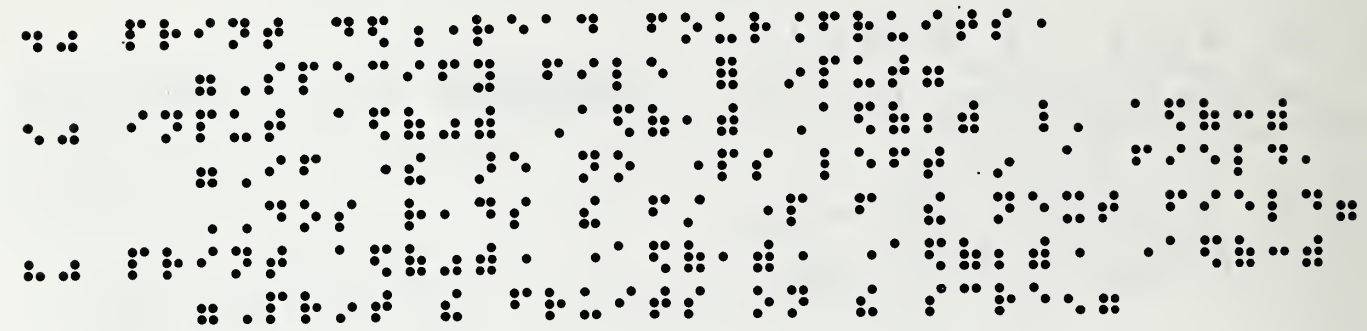
```
$ mv ch.* chapter.*      Doesn't work!
```

11.

```
$ cmp /usr/you/.profile /usr/you/.profile
$                                   No output; they're the same
$
0                                   Zero implies ran O.K.. files identical
$ cmp /usr/you/.profile /usr/mary/.profile
/usr/you/.profile /usr/mary/.profile differ: char 6, line 3
$ echo $?
1                                   Non-zero means files were different
$
```

12. Author's comments are enclosed in parentheses.

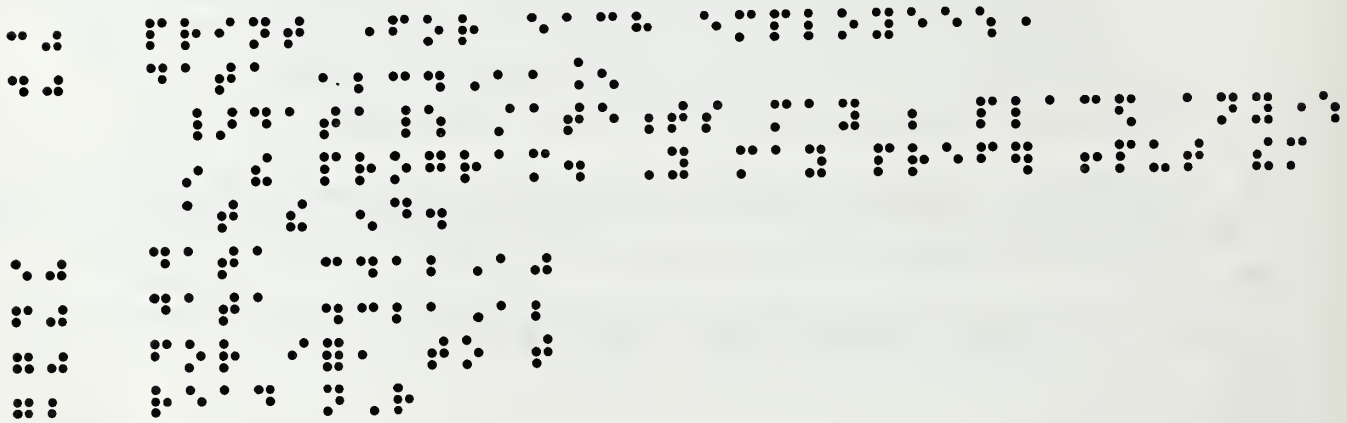
```
40 PRINT D$;"READ FOUR.FRUITS"
                              (Specify file for input)
50 INPUT A$(0),A$(1),A$(2), A$(3)
                              (If there are no parts left in a
                              field, DOS reads the first part
                              from the next field)
80 PRINT A$(0)" "A$(1)" "A$(2)" "A$(3)
                              (Print the fruits on the screen)
```



13. Author's comments are upper and lower case. Program lines are entirely upper case.

```
5  LET T=3
10  PRINT "THIS PROGRAM HAS DATA FOR"
20  PRINT T;" EMPLOYEES."
30  PRINT "FOR EACH EMPLOYEE:"
40  DATA 1234,11.5
50  DATA 3412,10
60  DATA 4321,12
70  FOR I=1 TO T
72  READ N,R
```

DATA statements may be placed anywhere in the program. You may prefer to put them at the end.



Occasionally, both types of comments - those that are included in the program lines, and those that are inserted by the author - appear. Transcribe according to the rules for each type.

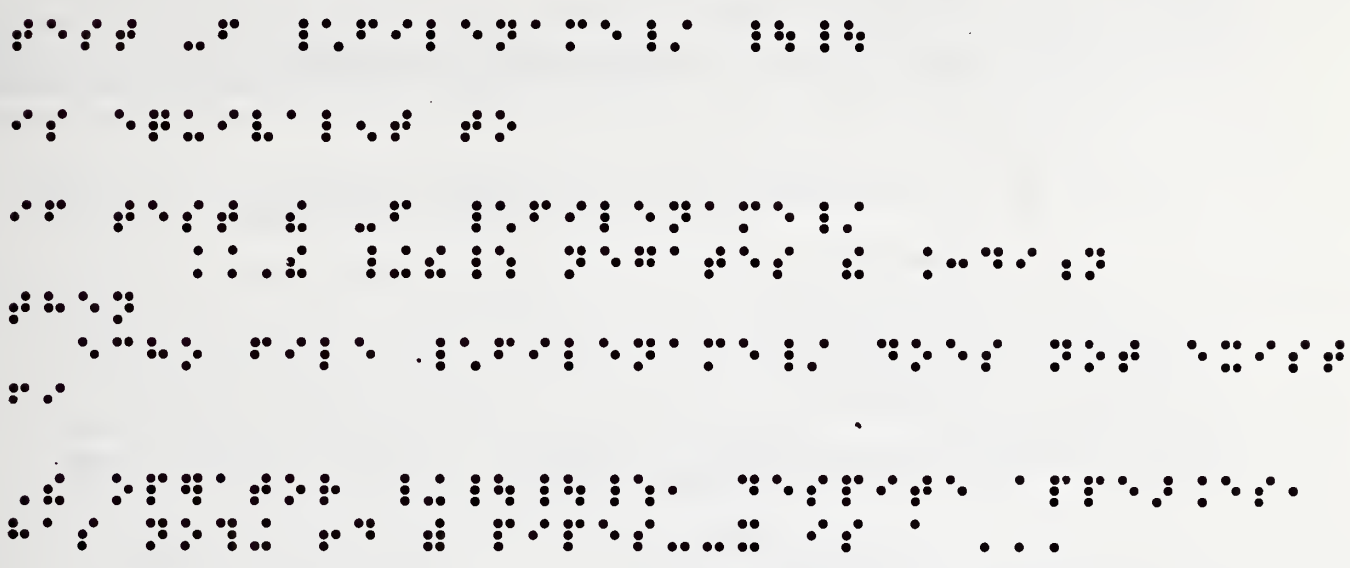
16.

```
test -f filename ||
```

is equivalent to

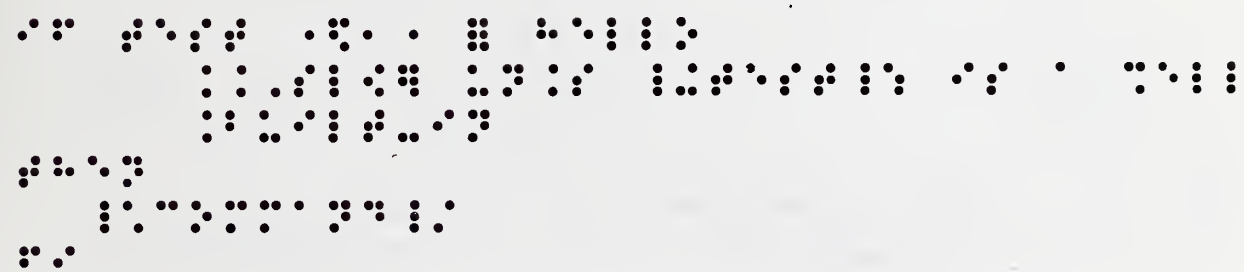
```
if test ! -f filename                    The ! negates the condition
then
        echo file filename does not exist
fi
```

The operator ||, despite appearances, has nothing to do with pipes — it is a



17.

```
if test "$1" = hello          Slower unless test is a shell built-in
then
        command
fi
```



XXII INDENTATION
(Section VII - CBC)

The rules for indentation apply to programs, program segments, or program-like material that is set off from the body of the text by blank lines.

Although the rules for indentation are specific and will apply to most computer-related material that is being used at this time, exceptions will need to be made for unusual circumstances. The CBC states (Section II - C.) that, when in doubt, assume spacing of vertical or horizontal text is significant. It is also very helpful if you can consult with the reader, or the teacher of a student reader, or a computer programming expert of your acquaintance in order to determine the significance of the spacing.

Here are some examples in which spacing would be considered to be significant.

1.

```
C
C       SET THE TABLES TO 0.0 BEFORE PROCESSING
C
      DO 10 I = 1,N
      A(I) = 0.0
      B(I) = 0.0
      C(I) = 0.0
   10 CONTINUE
```

2. Consider the following program:

```
          X = 1234567.
          DO 10 I = 1,1000
       10 X = X T .01
          WRITE(3,101) X
      101 FORMAT(' X=',F9.0)
          CALL EXIT
          END
```

3.

```
case $# in
0)      echo 'Usage: which command' 1>&2; exit 2
esac
for i in `echo $PATH | sed 's/^:/.:/
                            s/::/:.:/g
                            s/:$/:./
                            s/:/ /g'`
do
```

4.

```
$ cal
usage: cal [month] year
$ cal october 1983
Bad argument
$ cal 10 1983
   October 1983
 S  M Tu  W Th  F  S
                1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
$
```

XXIII. STRUCTURING THE TEXT

Due to the rapidly developing art and technology of computers, their software, and applications, the BANA Computer Braille Code committee felt that any rigidly structured code would soon be outdated. Consequently, there was a deliberate attempt to build flexibility into the Computer Braille Code. This very flexibility places a strong mandate upon the structurer of a transcription. A thorough search of the entire text is absolutely essential. The legibility of the text will depend to a much greater extent upon the skills of the person doing the structuring than has been true previously with braille transcriptions.

The flexibility of the code leaves several areas open to a decision by the transcriber. In these areas more than one method may be "correct". Different interpretations of meaning may lead to inconsistencies from one text to another. This is unavoidable, but being consistent throughout a text is mandatory.

A. Following is an outline for structuring a text and suggestions for making the necessary decisions.

1. Deciding on the primary code

Most textbooks and manuals that are used for computer-related materials will require the use of the Literary and Textbook Format Code. However, if a book has many mathematical expressions throughout, it may be better transcribed in Nemeth Code as the primary code. Mathematics texts that have "computer corners", or some other-named computer-related sections, must use Nemeth Code as the primary code.

2. Designating type styles

The majority of books dealing with computer-related materials will make use of various different type styles or type fonts to specify different areas of computer notation. These different forms of type are almost always explained or pointed out in the text.

The most common uses for variant type styles are

a. to designated programs or program lines
b. to designate material that must be entered by the user
c. to designate comments by the author that are included in program lines
d. to designate screen displays

The type style used for programs and program lines should be the base or primary computer notation. Use the Computer Braille Code indicators when this type style is in embedded text as indicated the the rules of the Computer Braille Code. (Other indicators may be necessary to introduce this type style also. See Section IX.) When a second type style is used in conjunction with the basic style, interspersed with program lines or by itself, and designated for some use other than programs, use the Emphasis Indicators for this style.

Occasionally a third type style will be used for another purpose. Use the Transcriber's Option indicator.

Note: If italics are used for literary emphasis - use the primary code. Make a list of the various type styles and the indicators you will be using for them.

In the event that only one type style is used in a text, you will have to decide which indicators to use by context. For example, upper-case words will usually require indicators. Make your decision based on the guidelines and examples in the code and this presentation.

3. Deciding whether to use the Caps Lock indicator

If the letters in programs are all upper-case, you will not need to use the Caps Lock indicators. (Remember to include a transcriber's note at the beginning of the work.) This also holds true if virtually all letters are upper-case with only an occasional notation in lower-case.

4. Deciding on shape symbols

Make a list of any shapes that are not represented by computer braille code symbols. Decide whether these shapes are part of computer notation by looking for them in programs or program lines. Assign letter combinations for each shape. The text often explains what a shape stands for - or use letters that represent the shape as it looks to you. This list of shapes must be included on a "Special Symbols" page at the beginning of the book.

If you are a Nemeth transcriber, you may use Nemeth symbols for shapes.

Use symbols as specified in the primary (literary) code for shapes that are not computer notation.

5. Computer Braille Code symbols

Make a list of the graphic symbols used in the text (those symbols other than letters or numbers). Some texts will have a list of these symbols. The list, if present, could be in the first chapter or two, in an appendix, in a glossary, on the inside cover, or as the first listings in an index. If the text does not have a list, check through the programs and then through the pages of the book.

Compare your list of symbols with those in the code. Symbols on your list that are not included in the code may be replaced by symbols in the code that are not on your list. Or the Transcriber's Option symbols may be assigned to two symbols that are not included in the code. (Remember that you must not assign a Transcriber's Option symbol to both a type style and a graphic symbol.) Include a transcriber's note at the beginning of the work for any symbol substitutions.

6. Format

Look over programs and other displayed material.

Decide if programs should be transcribed using the "indentation" rules of the code. The text may explain any particular format that is required for programs. If there is no explanation in the text, use the guidelines and

examples in the code and this presentation to help you make your decision.

B. Following are some additional suggestions for making decisions while structuring the text or, after you have begun transcribing, in the areas where more than one construction is permissible.

1. Computer Notation

It is not always completely clear as to what is computer notation, even when different type styles are used. Some texts may use italics in programs and also for emphasis in literary passages. Emphasis indicators or Transcriber's Option symbols may be required in the programs and literary italic signs in the literary passages. Some texts use upper-case letters in programs and other computer notation where the Caps Lock Indicators may or may not be required (see below), and also use upper-case letters for the keys of the keyboard such as RETURN, DELETE, etc., or for emphasis in literary passages, where the literary double cap sign should be used.

It can, in most cases, be determined whether to use computer notation or literary notation by context, code guidelines, or common sense. However, if the distinction is not really clear and the notation could readily be transcribed with literary meaning, the use of literary notation is preferable in order to make the reader's task easier. Just remember than consistency is most important.

2. Caps Lock Indicators

When it has been decided during text structuring that upper-case letters are used predominantly in the text for computer notation, the transcriber may choose not to use the Caps Lock indicators for such notation. (Include a transcriber's note.) However, a text that is only "predomonantly" (not "exclusively") upper-case will contain some lower-case letters within some portions of computer notation.

If lower-case letters are always set off by some particular symbols - such as those used to enclose "comments", or byparentheses which are not otherwise used in programs - consider transcribing these lower-case letters without indicators. A transcriber's note at the beginning of the work should explain this procedure.

If the lower-case letters are not distinguishable in any particular way, the Caps Lock, Shift, and Caps Release indicators should be used as illustrated in Section V.

3. Caps Release Indicator

When computer notation that begins with the Caps Lock indicator is followed, without an intervening space, by lower-case computer notation, the Caps Release Indicator is used to terminate upper-case. The Caps Release can be placed at any point following the upper-case letters and preceding the lower-case letters. Each individual notation must be judged separately and the Caps Release placed in what seems to be the most logical position.

When an upper-case computer "word" is followed by lower-case letters to make the computer word descriptive - for example, CLOSEs, WITHed. BEGIN-block, etc. - it would be technically correct to use either the Caps Release or the Termination indicator to end upper-case. However, it would seem that the Termination Indicator would represent the situation more precisely in this case.

4. Mathematical Expressions

Technically any expression that contains numbers or letters and mathematical signs could be considered mathematical. Unless the primary code being used is the Nemeth Code, however, mathematical expressions which describe computer processes and which contain only the mathematical signs that are included in the list of computer code symbols should be transcribed in computer notation.

If you are not a Nemeth transcriber, consult one who is for help with mathematical expressions that contain signs which are not included in the computer code symbols. Or contact your Computer Code Specialist for help.

NOTE: Be on the lookout for any comments in the text that might be helpful in your transcription. For instance, the following kinds of comments could help make the transcription of a program easier if you were concerned about spacing.

out what, exactly, it was supposed to do. In PASCAL, spaces and new lines are not significant, except when they are used to separate individual words, or symbols, within the program. This is to allow the programmer to lay out his programs in the most appropriate way to show the overall structure, and to make it as clear and as readable as possible.

Another significant aspect of APL is that spacing is very important. That is, the number of spaces between symbols and numbers has to be preserved, and thus

XXIV SUGGESTIONS

SCREEN DISPLAYS - Treat the same as other text (primary code except when computer notation is required) even though they may be in a variant type style.

FLOW CHARTS - Use the symbols and format from the 1972 Provisional Computer Code. Symbols will need to be listed on the Special Symbols Page. Put them together under a separate heading "Flowchart Symbols".

SYNTAX DIAGRAMS - These have to be studied. Some can be done using flow chart format. Some can be done using the textbook/literary outline format.

ERRATA

Page 3, Example #1 -- Comma after RUN should be a literary comma, dot 2

Page 10, Example #1a -- Omit the symbol dots 3-4 at the end of the line.

Page 28, Second paragraph --"Capts" should be "Caps"

Page 33, Fxample #2 -- "wh" sign should be "gh" sign

Page 43, Example #4 -- braille does not correspond to the example. There should not be emphasis indicators in this example.

Page 55, Example #4 -- Semicolons in print should be colons. Braille is correct.

Page 59, Example #2a -- DN should be ND in RANDom
Example #2b -- 6 should be 9 in line 3

Page 63, Example #3 -- Yl should be dots 4-5-6 Y instead of dot 6 Y

Page 72, Fxample #2 -- Second line should have a minus sign between the Half-line Shift Down and the 3

Page 76, Example #1 -- Braille inadvertently left out.

Page 78, last line of first braille example -- "indicates" should have s instead of t for the last letter.

Page 82, Example #3a -- should be additional instead of addition

Page 84, Example #8d -- Delete dots 3-4 at end of braille example

Pages 87 -- Should be a period after RETURN (in key shape)

Page 96, III. B. -- should read "Computer Braille Code Indicator"

Page 100 -- Delete the first three lines--they are a repeat from the previous page.